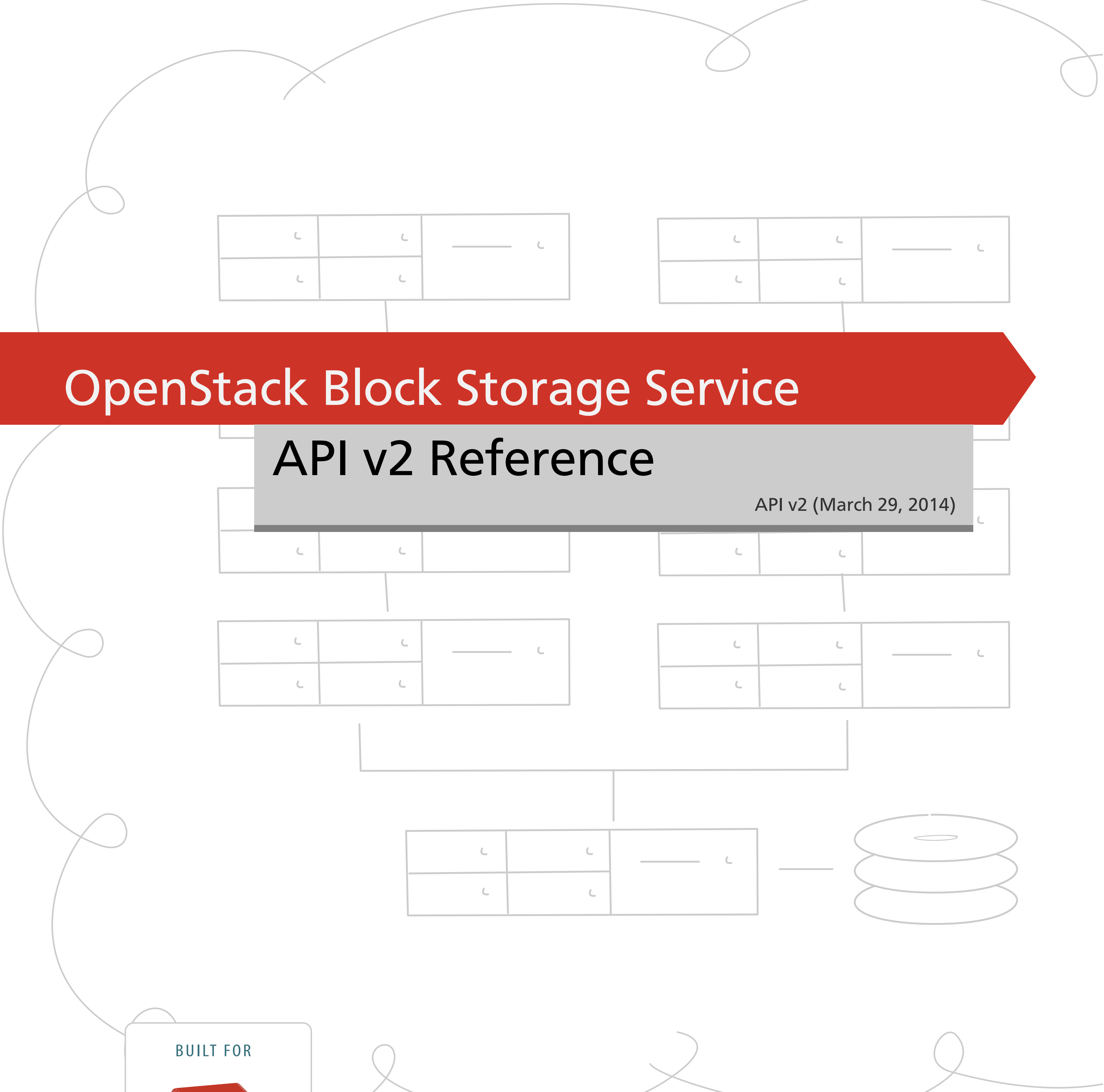
OpenStack Block Storage Service

API v2 Reference

API v2 (March 29, 2014)



BUILT FOR

openstack™

CLOUD SOFTWARE

openstack™

docs.openstack.org

OpenStack Block Storage Service API v2 Reference

API v2 (2014-03-29)

This document is for software developers who develop applications by using the OpenStack Block Storage Service Application Programming Interface (API).

Table of Contents

List of Tables

List of Examples

Preface

OpenStack Block Storage provides volume management with the OpenStack Compute service.

This document describes the features available with the Block Storage API v2.0.

We welcome feedback, comments and bug reports at bugs.launchpad.net/Cinder.

Intended audience

This guide assists software developers who develop applications by using the Block Storage API v2.0. It assumes the reader has a general understanding of storage and is familiar with:

- ReSTful web services
- HTTP/1.1 conventions
- JSON and/or XML data serialization formats

Additional resources

You can download the latest API-related documents from docs.openstack.org/api/.

This API uses standard HTTP 1.1 response codes as documented at: www.w3.org/Protocols/rfc2616/rfc2616-sec10.html.

1. Overview

OpenStack Block Storage Service is a block-level storage solution that enables you to:

- Mount drives to OpenStack Cloud Servers™ to scale storage without paying for more compute resources.
- Use high performance storage to serve database or I/O-intensive applications.

You interact with Block Storage programmatically through the Block Storage API as described in this guide.

Note

- OpenStack Block Storage Service is an add-on feature to OpenStack Nova Compute in Folsom versions and earlier.
- Block Storage is multi-tenant rather than dedicated.
- Block Storage allows you to create snapshots that you can save, list, and restore.
- Block Storage allows you to create backups of your volumes to Object Storage for archival and disaster recovery purposes. These backups can be subsequently restored to the same volume or new volumes.

Glossary

To use the Block Storage API effectively, you must understand several key concepts:

- **Volume**

  A detachable block storage device. You can think of it as a USB hard drive. It can only be attached to one instance at a time.

- **Volume type**

  A type of a block storage volume. You can define whatever types work best for you, such as SATA, SCSCI, SSD, etc. These can be customized or defined by the OpenStack admin.

  You can also define extra_specs associated with your volume types. For instance, you could have a VolumeType=SATA, with extra_specs (RPM=10000, RAID-Level=5) . Extra_specs are defined and customized by the admin.

- **Snapshot**

  A point in time copy of the data contained in a volume.

- **Instance**

  A virtual machine (VM) that runs inside the cloud.

- **Backup**

  A full copy of a volume stored in an external service. The service can be configured. The only supported service for now is Object Storage. A backup can subsequently be restored from the external service to either the same volume that the backup was originally taken from, or to a new volume.

High-level task flow

Procedure 1.1. To create and attach a volume

1. You create a volume.

   For example, you might create a 30 GB volume called `vol1`, as follows:

   ```
   $ cinder create --display-name vol1 30
   ```

   The command returns the `521752a6-acf6-4b2d-bc7a-119f9148cd8c` volume ID.

2. You attach that volume to a virtual machine (VM) with the `616fb98f-46ca-475e-917e-2563e5a8cd19` ID, as follows:

   For example:

   ```
   $ nova volume-attach 616fb98f-46ca-475e-917e-2563e5a8cd19 521752a6-
   acf6-4b2d-bc7a-119f9148cd8c /dev/vdb
   ```

2. General API information

The Block Storage API is implemented using a ReSTful web service interface. Like other OpenStack projects, Block Storage shares a common token-based authentication system that allows access between products and services.

Note

All requests to authenticate against and operate the service are performed using SSL over HTTP (HTTPS) on TCP port 443.

Authentication

You can use cURL to try the authentication process in two steps: get a token, and send the token to a service.

1. Get an authentication token by providing your user name and either your API key or your password. Here are examples of both approaches:

   You can request a token by providing your user name and your password.

   ```
   $ curl -X POST https://localhost:5000/v2.0/tokens -d '{"auth":
   {"passwordCredentials":{"username": "joecool", "password":"coolword"},
    "tenantId":"5"}}' -H 'Content-type: application/json'
   ```

   Successful authentication returns a token which you can use as evidence that your identity has already been authenticated. To use the token, pass it to other services as an `X-Auth-Token` header.

   Authentication also returns a service catalog, listing the endpoints you can use for Cloud services.

2. Use the authentication token to send a GET to a service you would like to use.

Authentication tokens are typically valid for 24 hours. Applications should be designed to re-authenticate after receiving a 401 (Unauthorized) response from a service endpoint.

Important

If you programmatically parse an authentication response, be aware that service names are stable for the life of the particular service and can be used as keys. You should also be aware that a user's service catalog can include multiple uniquely-named services that perform similar functions.

Request and response types

The Block Storage API supports both the JSON and XML data serialization formats. The request format is specified using the `Content-Type` header and is required for calls that have a request body. The response format can be specified in requests either by using the `Accept` header or by adding an `.xml` or `.json` extension to the request URI. Note that it is possible for a response to be serialized using a format different from the request. If no response format is specified, JSON is the default. If conflicting formats are specified using both an `Accept` header and a query extension, the query extension takes precedence.

Table 2.1. Response formats

Format	Accept Header	Query Extension	Default
JSON	application/json	.json	Yes
XML	application/xml	.xml	No

In the request example below, notice that *`Content-Type`* is set to *`application/json`*, but *`application/xml`* is requested via the *`Accept`* header:

Example 2.1. Request with headers: Get volume types

```
GET /v2/441446/types HTTP/1.1
Host: dfw.blockstorage.api.openstackcloud.com
X-Auth-Token: eaaafd18-0fed-4b3a-81b4-663c99ec1cbb
Accept: application/xml
```

An XML response format is returned:

Example 2.2. Response with headers

```
HTTP/1.1 200 OK
Date: Fri, 20 Jul 2012 20:32:13 GMT
Content-Length: 187
Content-Type: application/xml
X-Compute-Request-Id: req-8e0295cd-a283-46e4-96da-cae05cbfd1c7

<?xml version='1.0' encoding='UTF-8'?>
  <volume_types>
      <volume_type id="1" name="SATA">
          <extra_specs/>
      </volume_type>
      <volume_type id="2" name="SSD">
          <extra_specs/>
      </volume_type>
  </volume_types>
```

Limits

All accounts, by default, have a preconfigured set of thresholds (or limits) to manage capacity and prevent abuse of the system. The system recognizes two kinds of limits: *rate limits* and *absolute limits*. Rate limits are thresholds that are reset after a certain amount of time passes. Absolute limits are fixed.

Absolute limits

The following table shows the absolute limits:

Table 2.2. Absolute limits

Name	Description	Limit
Block Storage	Maximum amount of block storage	1 TB

Date and time format

The Block Storage Service uses an ISO-8601 compliant date format for the display and consumption of date time values.

Example 2.3. DB service date and time format

```
yyyy-MM-dd'T'HH:mm:ss.SSSZ
```

May 19th, 2011 at 8:07:08 AM, GMT-5 has the following format:

```
2011-05-19T08:07:08-05:00
```

The following table describes the date time format codes:

Table 2.3. Date time format codes

Code	Description
yyyy	Four digit year
MM	Two digit month
dd	Two digit day of month
T	Separator for date time
HH	Two digit hour of day (00-23)
mm	Two digit minutes of hour
ss	Two digit seconds of the minute
SSS	Three digit milliseconds of the second
Z	RFC-822 timezone

Faults

When an error occurs, the Block Storage Service returns a fault object containing an HTTP error response code that denotes the type of error. The response body returns additional information about the fault.

The following table lists possible fault types with their associated error codes and descriptions.

Fault type	Associated error code	Description
`badRequest`	400	The user request contains one or more errors.
`unauthorized`	401	The supplied token is not authorized to access the resources, either it's expired or invalid.
`forbidden`	403	Access to the requested resource was denied.

Fault type	Associated error code	Description
`itemNotFound`	404	The back-end services did not find anything matching the Request-URI.
`badMethod`	405	The request method is not allowed for this resource.
`overLimit`	413	Either the number of entities in the request is larger than allowed limits, or the user has exceeded allowable request rate limits. See the `details` element for more specifics. Contact support if you think you need higher request rate limits.
`badMediaType`	415	The requested content type is not supported by this service.
`unprocessableEntity`	422	The requested resource could not be processed on at the moment.
`instanceFault`	500	This is a generic server error and the message contains the reason for the error. This error could wrap several error messages and is a catch all.
`notImplemented`	501	The requested method or resource is not implemented.
`serviceUnavailable`	503	The Block Storage Service is not available.

The following two `instanceFault` examples show errors when the server has erred or cannot perform the requested operation:

Example 2.4. Example instanceFault response: XML

```
HTTP/1.1 500 Internal Server Error
Content-Type: application/xml
Content-Length: 121
Date: Mon, 28 Nov 2011 18:19:37 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<instanceFault code="500"
    xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content">
    <message> The server has either erred or is incapable of
        performing the requested operation. </message>
</instanceFault>
```

Example 2.5. Example fault response: JSON

```
HTTP/1.1 500 Internal Server Error
Content-Length: 120
Content-Type: application/json; charset=UTF-8
Date: Tue, 29 Nov 2011 00:33:48 GMT
```

```
{
   "instanceFault":{
      "code":500,
      "message":"The server has either erred or is incapable of performing the
 requested operation."
   }
}
```

The error code (`code`) is returned in the body of the response for convenience. The `message` element returns a human-readable message that is appropriate for display to the end user. The `details` element is optional and may contain information that is useful for tracking down an error, such as a stack trace. The `details` element may or may not be appropriate for display to an end user, depending on the role and experience of the end user.

The fault's root element (for example, `instanceFault`) may change depending on the type of error.

The following two badRequest examples show errors when the volume size is invalid:

Example 2.6. Example badRequest fault on volume size errors: XML

```
HTTP/1.1 400 None
Content-Type: application/xml
Content-Length: 121
Date: Mon, 28 Nov 2011 18:19:37 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<badRequest code="400"
    xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content">
    <message> Volume 'size' needs to be a positive integer value, -1.0
        cannot be accepted. </message>
</badRequest>
```

Example 2.7. Example badRequest fault on volume size errors: JSON

```
HTTP/1.1 400 None
Content-Length: 120
Content-Type: application/json; charset=UTF-8
Date: Tue, 29 Nov 2011 00:33:48 GMT
```

```
{
   "badRequest":{
      "code":400,
      "message":"Volume 'size' needs to be a positive integer value, -1.0
 cannot be accepted."
   }
}
```

The next two examples show itemNotFound errors:

Example 2.8. Example itemNotFound fault: XML

```
HTTP/1.1 404 Not Found
Content-Length: 147
Content-Type: application/xml; charset=UTF-8
Date: Mon, 28 Nov 2011 19:50:15 GMT
```

```
<?xml version="1.0" encoding="UTF-8"?>
<itemNotFound code="404"
    xmlns="http://docs.openstack.org/api/openstack-block-storage/2.0/content">
    <message> The resource could not be found. </message>
</itemNotFound>
```

Example 2.9. Example itemNotFound fault: JSON

```
HTTP/1.1 404 Not Found
Content-Length: 78
Content-Type: application/json; charset=UTF-8
Date: Tue, 29 Nov 2011 00:35:24 GMT
```

```
{
   "itemNotFound":{
      "code":404,
      "message":"The resource could not be found."
   }
}
```

3. API operations

Volumes

A volume is a detachable block storage device. You can think of it as a USB hard drive. You can attach a volume to one instance at a time.

When you create, list, or delete volumes, these status values are possible:

Status	Description
creating	The volume is being created.
available	The volume is ready to be attached to an instance.
attaching	The volume is attaching to an instance.
in-use	The volume is attached to an instance.
deleting	The volume is being deleted.
error	An error occurred during volume creation.
error_deleting	An error occurred during volume deletion.
backing-up	The volume is being backed up.
restoring-backup	A backup is being restored to the volume.
error_restoring	An error occurred during backup restoration to a volume.

Method	URI	Description
POST	`/v2/{tenant_id}/volumes`	Creates a volume.
GET	`/v2/{tenant_id}/volumes`	Lists summary information for all Block Storage volumes that the tenant who submits the request can access.
GET	`/v2/{tenant_id}/volumes/detail`	Lists detailed information for all Block Storage volumes that the tenant who submits the request can access.
GET	`/v2/{tenant_id}/volumes/{volume_id}`	Shows information about a specified volume.
PUT	`/v2/{tenant_id}/volumes/{volume_id}{?display_description, display_name}`	Updates a volume.
DELETE	`/v2/{tenant_id}/volumes/{volume_id}`	Deletes a specified volume.

Create volume

Method	URI	Description
POST	`/v2/{tenant_id}/volumes`	Creates a volume.

To create a bootable volume, include the image ID and set the `bootable` flag to `true` in the request body.

Normal response codes: 202

Request

This table shows the URI parameters for the create volume request:

Name	Type	Description
`{tenant_id}`	String	The unique identifier of the tenant or account.

Example 3.1. Create volume: JSON request

```
{
   "volume":{
      "availability_zone":null,
      "source_volid":null,
      "display_description":null,
      "snapshot_id":null,
      "size":10,
      "display_name":"my_volume",
      "imageRef":null,
      "volume_type":null,
      "metadata":{

      }
   }
}
```

Example 3.2. Create volume: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<volume
    xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
    display_name="vol-001" display_description="Another volume."
    size="2"/>
```

This operation does not require a request body.

Response

Example 3.3. Create volume: JSON response

```
{
   "volume":{
      "status":"creating",
      "display_name":"my_volume",
      "attachments":[
      ],
      "availability_zone":"nova",
      "bootable":"false",
      "created_at":"2014-02-21T19:52:04.949734",
      "display_description":null,
      "volume_type":"None",
      "snapshot_id":null,
      "source_volid":null,
      "metadata":{

      },
      "id":"93c2e2aa-7744-4fd6-a31a-80c4726b08d7",
      "size":10
   }
}
```

Example 3.4. Create volume: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<volume xmlns:atom="http://www.w3.org/2005/Atom"
    xmlns="http://docs.openstack.org/volume/api/v1" status="creating"
    display_name="vol-001" availability_zone="nova" bootable="false"
    created_at="2014-02-21 20:18:33.122452"
    display_description="Another volume." volume_type="None"
    snapshot_id="None" source_volid="None"
    id="83960a54-8dad-4fd8-bc41-33c71e098e04" size="2">
    <attachments/>
    <metadata/>
</volume>
```

This operation does not return a response body.

List volumes

Method	URI	Description
GET	`/v2/{tenant_id}/volumes`	Lists summary information for all Block Storage volumes that the tenant who submits the request can access.

Normal response codes: 200

Request

This table shows the URI parameters for the list volumes request:

Name	Type	Description
`{tenant_id}`	String	The unique identifier of the tenant or account.

This operation does not require a request body.

Response

Example 3.5. List volumes: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<volumes xmlns:atom="http://www.w3.org/2005/Atom"
    xmlns="http://docs.openstack.org/api/openstack-block-storage/2.0/content">
    <volume name="vol-004" id="45baf976-c20a-4894-a7c3-c94b7376bf55">
        <attachments/>
        <metadata/>
    </volume>
    <volume name="vol-003" id="5aa119a8-d25b-45a7-8d1b-88e127885635">
        <attachments/>
        <metadata/>
    </volume>
</volumes>
```

Example 3.6. List volumes: JSON response

```
{
    "volumes": [
        {
            "id": "45baf976-c20a-4894-a7c3-c94b7376bf55",
            "links": [
                {
                    "href": "http://localhost:8776/v2/
0c2eba2c5af04d3f9e9d0d410b371fde/volumes/45baf976-c20a-4894-a7c3-
c94b7376bf55",
                    "rel": "self"
                },
                {
                    "href": "http://localhost:8776/
0c2eba2c5af04d3f9e9d0d410b371fde/volumes/45baf976-c20a-4894-a7c3-
c94b7376bf55",
                    "rel": "bookmark"
                }
            ],
            "name": "vol-004"
        },
        {
            "id": "5aa119a8-d25b-45a7-8d1b-88e127885635",
            "links": [
                {
                    "href": "http://localhost:8776/
v2/0c2eba2c5af04d3f9e9d0d410b371fde/volumes/5aa119a8-
d25b-45a7-8d1b-88e127885635",
                    "rel": "self"
                },
                {
                    "href": "http://localhost:8776/
0c2eba2c5af04d3f9e9d0d410b371fde/volumes/5aa119a8-
d25b-45a7-8d1b-88e127885635",
                    "rel": "bookmark"
                }
            ],
            "name": "vol-003"
        }
    ]
}
```

This operation does not return a response body.

List volumes (detailed)

Method	URI	Description
GET	`/v2/{tenant_id}/volumes/detail`	Lists detailed information for all Block Storage volumes that the tenant who submits the request can access.

Normal response codes: 200

Request

This table shows the URI parameters for the list volumes (detailed) request:

Name	Type	Description
`{tenant_id}`	String	The unique identifier of the tenant or account.

This operation does not require a request body.

Response

Example 3.7. List volumes (detailed): XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<volumes
    xmlns:os-vol-image-meta="http://docs.openstack.org/openstack-block-
storage/2.0/content/Volume_Image_Metadata.html"
    xmlns:os-vol-tenant-attr="http://docs.openstack.org/openstack-block-
storage/2.0/content/Volume_Tenant_Attribute.html"
    xmlns:os-vol-host-attr="http://docs.openstack.org/openstack-block-storage/
2.0/content/Volume_Host_Attribute.html"
    xmlns:atom="http://www.w3.org/2005/Atom"
    xmlns="http://docs.openstack.org/api/openstack-block-storage/2.0/content">
    <volume status="available" name="vol-004" availability_zone="nova"
        created_at="2013-02-25 06:36:28" description="Another volume."
        volume_type="None" source_volid="None" snapshot_id="None"
        id="45baf976-c20a-4894-a7c3-c94b7376bf55" size="1"
        os-vol-tenant-attr:tenant_id="0c2eba2c5af04d3f9e9d0d410b371fde"
        os-vol-host-attr:host="ip-10-168-107-25">
        <attachments/>
        <metadata>
            <meta key="contents">junk</meta>
        </metadata>
    </volume>
    <volume status="available" name="vol-003" availability_zone="nova"
        created_at="2013-02-25 02:40:21"
        description="This is yet, another volume." volume_type="None"
        source_volid="None" snapshot_id="None"
        id="5aa119a8-d25b-45a7-8d1b-88e127885635" size="1"
        os-vol-tenant-attr:tenant_id="0c2eba2c5af04d3f9e9d0d410b371fde"
        os-vol-host-attr:host="ip-10-168-107-25">
        <attachments/>
        <metadata>
            <meta key="contents">not junk</meta>
        </metadata>
    </volume>
```

```
</volumes>
```

Example 3.8. List volumes (detailed): JSON response

```
{
   "volumes":[
      {
         "status":"available",
         "attachments":[

         ],
         "links":[
            {
               "href":"http://localhost:8776/v2/
0c2eba2c5af04d3f9e9d0d410b371fde/volumes/45baf976-c20a-4894-a7c3-
c94b7376bf55",
               "rel":"self"
            },
            {
               "href":"http://localhost:8776/0c2eba2c5af04d3f9e9d0d410b371fde/
volumes/45baf976-c20a-4894-a7c3-c94b7376bf55",
               "rel":"bookmark"
            }
         ],
         "availability_zone":"nova",
         "os-vol-host-attr:host":"ip-10-168-107-25",
         "source_volid":null,
         "snapshot_id":null,
         "id":"45baf976-c20a-4894-a7c3-c94b7376bf55",
         "description":"Another volume.",
         "name":"vol-004",
         "created_at":"2013-02-25T06:36:28.000000",
         "volume_type":"None",
         "os-vol-tenant-attr:tenant_id":"0c2eba2c5af04d3f9e9d0d410b371fde",
         "size":1,
         "metadata":{
            "contents":"junk"
         }
      },
      {
         "status":"available",
         "attachments":[

         ],
         "links":[
            {
               "href":"http://localhost:8776/v2/
0c2eba2c5af04d3f9e9d0d410b371fde/volumes/5aa119a8-
d25b-45a7-8d1b-88e127885635",
               "rel":"self"
            },
            {
               "href":"http://localhost:8776/0c2eba2c5af04d3f9e9d0d410b371fde/
volumes/5aa119a8-d25b-45a7-8d1b-88e127885635",
               "rel":"bookmark"
            }
         ],
         "availability_zone":"nova",
         "os-vol-host-attr:host":"ip-10-168-107-25",
         "source_volid":null,
         "snapshot_id":null,
         "id":"5aa119a8-d25b-45a7-8d1b-88e127885635",
         "description":"This is yet, another volume.",
         "name":"vol-003",
         "created_at":"2013-02-25T02:40:21.000000",
         "volume_type":"None",
         "os-vol-tenant-attr:tenant_id":"0c2eba2c5af04d3f9e9d0d410b371fde",
         "size":1,
         "metadata":{
            "contents":"not junk"
         }
      }
   ]
}
```

This operation does not return a response body.

Show volume information

Method	URI	Description
GET	`/v2/{tenant_id}/volumes/{volume_id}`	Shows information about a specified volume.

Normal response codes: 200

Request

This table shows the URI parameters for the show volume information request:

Name	Type	Description
`{tenant_id}`	String	The unique identifier of the tenant or account.
`{volume_id}`	UUID	The unique identifier of an existing volume.

This operation does not require a request body.

Response

Example 3.9. Show volume information: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<volume
    xmlns:os-vol-image-meta="http://docs.openstack.org/openstack-block-
storage/2.0/content/Volume_Image_Metadata.html"
    xmlns:os-vol-tenant-attr="http://docs.openstack.org/openstack-block-
storage/2.0/content/Volume_Tenant_Attribute.html"
    xmlns:os-vol-host-attr="http://docs.openstack.org/openstack-block-storage/
2.0/content/Volume_Host_Attribute.html"
    xmlns:atom="http://www.w3.org/2005/Atom"
    xmlns="http://docs.openstack.org/api/openstack-block-storage/2.0/content"
    status="available" name="vol-003" availability_zone="nova"
    created_at="2013-02-25 02:40:21"
    description="This is yet, another volume." volume_type="None"
    source_volid="None" snapshot_id="None"
    id="5aa119a8-d25b-45a7-8d1b-88e127885635" size="1"
    os-vol-tenant-attr:tenant_id="0c2eba2c5af04d3f9e9d0d410b371fde"
    os-vol-host-attr:host="ip-10-168-107-25">
    <attachments/>
    <metadata>
        <meta key="contents">not junk</meta>
    </metadata>
</volume>
```

Example 3.10. Show volume information: JSON response

```
{
   "volume":{
      "status":"available",
      "attachments":[

      ],
      "links":[
         {
            "href":"http://localhost:8776/v2/0c2eba2c5af04d3f9e9d0d410b371fde/
volumes/5aa119a8-d25b-45a7-8d1b-88e127885635",
            "rel":"self"
         },
         {
            "href":"http://localhost:8776/0c2eba2c5af04d3f9e9d0d410b371fde/
volumes/5aa119a8-d25b-45a7-8d1b-88e127885635",
            "rel":"bookmark"
         }
      ],
      "availability_zone":"nova",
      "os-vol-host-attr:host":"ip-10-168-107-25",
      "source_volid":null,
      "snapshot_id":null,
      "id":"5aa119a8-d25b-45a7-8d1b-88e127885635",
      "description":"Super volume.",
      "name":"vol-002",
      "created_at":"2013-02-25T02:40:21.000000",
      "volume_type":"None",
      "os-vol-tenant-attr:tenant_id":"0c2eba2c5af04d3f9e9d0d410b371fde",
      "size":1,
      "metadata":{
         "contents":"not junk"
      }
   }
}
```

This operation does not return a response body.

Update volume

Method	URI	Description
PUT	`/v2/{tenant_id}/volumes/{volume_id}{?display_description,display_name}`	Updates a volume.

Normal response codes: 200

Request

This table shows the URI parameters for the update volume request:

Name	Type	Description
`{tenant_id}`	String	The unique identifier of the tenant or account.
`{volume_id}`	UUID	The unique identifier of an existing volume.

This table shows the query parameters for the update volume request:

Name	Type	Description
`display_description`	String *(Optional)*	A description of the volume.
`display_name`	String *(Optional)*	The name of the volume.

Example 3.11. Update volume: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshot
    xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
    display_name="vol-003" display_description="This is yet, another volume."/
>
```

Example 3.12. Update volume: JSON request

```
{
   "volume":{
      "display_name":"vol-003",
      "display_description":"This is yet, another volume."
   }
}
```

This operation does not require a request body.

Response

Example 3.13. Update volume: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<volume xmlns:atom="http://www.w3.org/2005/Atom"
    xmlns="http://docs.openstack.org/api/openstack-block-storage/2.0/content"
    status="available" display_name="vol-003" availability_zone="nova"
    created_at="2013-02-25 02:40:21"
    display_description="This is yet, another volume." volume_type="None"
    source_volid="None" snapshot_id="None"
    id="5aa119a8-d25b-45a7-8d1b-88e127885635" size="1">
    <attachments/>
    <metadata>
        <meta key="contents">not junk</meta>
    </metadata>
</volume>
```

Example 3.14. Update volume: JSON response

```
{
   "volume":{
      "status":"available",
      "attachments":[

      ],
      "links":[
         {
            "href":"http://localhost:8776/v2/0c2eba2c5af04d3f9e9d0d410b371fde/
volumes/5aa119a8-d25b-45a7-8d1b-88e127885635",
            "rel":"self"
         },
         {
            "href":"http://localhost:8776/0c2eba2c5af04d3f9e9d0d410b371fde/
volumes/5aa119a8-d25b-45a7-8d1b-88e127885635",
            "rel":"bookmark"
         }
      ],
      "availability_zone":"nova",
      "source_volid":null,
      "snapshot_id":null,
      "id":"5aa119a8-d25b-45a7-8d1b-88e127885635",
      "display_description":"This is yet, another volume.",
      "display_name":"vol-003",
      "created_at":"2013-02-25T02:40:21.000000",
      "volume_type":"None",
      "size":1,
      "metadata":{
         "contents":"not junk"
      }
   }
}
```

This operation does not return a response body.

Delete volume

Method	URI	Description
DELETE	`/v2/{tenant_id}/volumes/{volume_id}`	Deletes a specified volume.

Normal response codes: 202

Request

This table shows the URI parameters for the delete volume request:

Name	Type	Description
`{tenant_id}`	String	The unique identifier of the tenant or account.
`{volume_id}`	UUID	The unique identifier of an existing volume.

This operation does not require a request body.

Snapshots

A snapshot is a point in time copy of the data that a volume contains.

When you create, list, or delete snapshots, these status values are possible:

Status	Description
creating	The snapshot is being created.
available	The snapshot is ready to be used.
deleting	The snapshot is being deleted.
error	An error occurred during snapshot creation.
error_deleting	An error occurred during snapshot deletion.

Method	URI	Description
POST	`/v2/{tenant_id}/snapshots{?snapshot,volume_id,force,display_name,display_description}`	Creates a snapshot, which is a point-in-time copy of a volume. You can create a volume from the snapshot.
GET	`/v2/{tenant_id}/snapshots`	Lists summary information for all Block Storage snapshots that the tenant who submits the request can access.
GET	`/v2/{tenant_id}/snapshots/detail`	Lists detailed information for all Block Storage snapshots that the tenant who submits the request can access.
GET	`/v2/{tenant_id}/snapshots/{snapshot_id}`	Shows information for a specified snapshot.
PUT	`/v2/{tenant_id}/snapshots/{snapshot_id}{?display_description,display_name}`	Updates a specified snapshot.
DELETE	`/v2/{tenant_id}/snapshots/{snapshot_id}`	Deletes a specified snapshot.

Create snapshot

Method	URI	Description
POST	`/v2/{tenant_id}/snapshots {?snapshot,volume_id,force, display_name,display_description}`	Creates a snapshot, which is a point-in-time copy of a volume. You can create a volume from the snapshot.

Normal response codes: 202

Request

This table shows the URI parameters for the create snapshot request:

Name	Type	Description
`{tenant_id}`	String	The unique identifier of the tenant or account.

This table shows the query parameters for the create snapshot request:

Name	Type	Description
`snapshot`	String *(Required)*	A partial representation of a snapshot used in the creation process.
`volume_id`	String *(Required)*	To create a snapshot from an existing volume, specify the ID of the existing volume.
`force`	Boolean *(Optional)*	[True/False] Indicate whether to snapshot, even if the volume is attached. Default==False.
`display_name`	String *(Optional)*	Name of the snapshot. Default==None.
`display_description`	String *(Optional)*	Description of snapshot. Default==None.

Example 3.15. Create snapshot: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshot
    xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
    name="snap-001" description="Daily backup"
    volume_id="5aa119a8-d25b-45a7-8d1b-88e127885635" force="true"/>
```

Example 3.16. Create snapshot: JSON request

```
{
    "snapshot": {
        "name": "snap-001",
        "description": "Daily backup",
        "volume_id": "5aa119a8-d25b-45a7-8d1b-88e127885635",
        "force": true
     }
}
```

This operation does not require a request body.

Response

Example 3.17. Create snapshot: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<snapshot status="creating" description="Daily backup"
    created_at="2013-02-25T03:56:53.081642"
    volume_id="5aa119a8-d25b-45a7-8d1b-88e127885635" size="1"
    id="ffa9bc5e-1172-4021-acaf-cdcd78a9584d" name="snap-001">
    <metadata/>
</snapshot>
```

Example 3.18. Create snapshot: JSON response

```
{
   "snapshot":{
      "status":"creating",
      "description":"Daily backup",
      "created_at":"2013-02-25T03:56:53.081642",
      "metadata":{

      },
      "volume_id":"5aa119a8-d25b-45a7-8d1b-88e127885635",
      "size":1,
      "id":"ffa9bc5e-1172-4021-acaf-cdcd78a9584d",
      "name":"snap-001"
   }
}
```

This operation does not return a response body.

List snapshots

Method	URI	Description
GET	`/v2/{tenant_id}/snapshots`	Lists summary information for all Block Storage snapshots that the tenant who submits the request can access.

Normal response codes: 200

Request

This table shows the URI parameters for the list snapshots request:

Name	Type	Description
`{tenant_id}`	String	The unique identifier of the tenant or account.

This operation does not require a request body.

Response

Example 3.19. List snapshots: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<snapshots>
    <snapshot status="available" description="Very important"
        created_at="2013-02-25 04:13:17"
        volume_id="5aa119a8-d25b-45a7-8d1b-88e127885635" size="1"
        id="2bb856e1-b3d8-4432-a858-09e4ce939389" name="snap-001">
        <metadata/>
    </snapshot>
    <snapshot status="available" description="Weekly backup"
        created_at="2013-02-25 07:20:38"
        volume_id="806092e3-7551-4fff-a005-49016f4943b1" size="1"
        id="e820db06-58b5-439d-bac6-c01faa3f6499" name="snap-002">
        <metadata/>
    </snapshot>
</snapshots>
```

Example 3.20. List snapshots: JSON response

```
{
   "snapshots":[
      {
         "status":"available",
         "description":"Very important",
         "created_at":"2013-02-25T04:13:17.000000",
         "metadata":{

         },
         "volume_id":"5aa119a8-d25b-45a7-8d1b-88e127885635",
         "size":1,
         "id":"2bb856e1-b3d8-4432-a858-09e4ce939389",
         "name":"snap-001"
      },
      {
         "status":"available",
         "description":"Weekly backup",
         "created_at":"2013-02-25T07:20:38.000000",
         "metadata":{

         },
         "volume_id":"806092e3-7551-4fff-a005-49016f4943b1",
         "size":1,
         "id":"e820db06-58b5-439d-bac6-c01faa3f6499",
         "name":"snap-002"
      }
   ]
}
```

This operation does not return a response body.

List snapshots (detailed)

Method	URI	Description
GET	`/v2/{tenant_id}/snapshots/detail`	Lists detailed information for all Block Storage snapshots that the tenant who submits the request can access.

Normal response codes: 200

Request

This table shows the URI parameters for the list snapshots (detailed) request:

Name	Type	Description
`{tenant_id}`	String	The unique identifier of the tenant or account.

This operation does not require a request body.

Response

Example 3.21. List snapshots (detailed): XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<snapshots
    xmlns:os-extended-snapshot-attributes="http://docs.openstack.org/
openstack-block-storage/2.0/content/Extended_Snapshot_Attributes.html">
    <snapshot status="available" description="Daily backup"
        created_at="2013-02-25 07:30:12"
        volume_id="5aa119a8-d25b-45a7-8d1b-88e127885635" size="30"
        id="43f20e0e-2c2c-4770-9d4e-c3d769ae5470" name="snap-001"
        os-extended-snapshot-attributes:project_id=
"0c2eba2c5af04d3f9e9d0d410b371fde"
        os-extended-snapshot-attributes:progress="100%">
        <metadata/>
    </snapshot>
    <snapshot status="available" description="Weekly backup"
        created_at="2013-02-25 07:20:38"
        volume_id="806092e3-7551-4fff-a005-49016f4943b1" size="1"
        id="e820db06-58b5-439d-bac6-c01faa3f6499" name="snap-002"
        os-extended-snapshot-attributes:project_id=
"0c2eba2c5af04d3f9e9d0d410b371fde"
        os-extended-snapshot-attributes:progress="100%">
        <metadata/>
    </snapshot>
</snapshots>
```

Example 3.22. List snapshots (detailed): JSON response

```
{
   "snapshots":[
      {
         "status":"available",
         "os-extended-snapshot-attributes:progress":"100%",
         "description":"Daily backup",
         "created_at":"2013-02-25T07:30:12.000000",
         "metadata":{
         },
         "volume_id":"5aa119a8-d25b-45a7-8d1b-88e127885635",
         "os-extended-snapshot-
attributes:project_id":"0c2eba2c5af04d3f9e9d0d410b371fde",
         "size":30,
         "id":"43f20e0e-2c2c-4770-9d4e-c3d769ae5470",
         "name":"snap-001"
      },
      {
         "status":"available",
         "os-extended-snapshot-attributes:progress":"100%",
         "description":"Weekly backup",
         "created_at":"2013-02-25T07:20:38.000000",
         "metadata":{

         },
         "volume_id":"806092e3-7551-4fff-a005-49016f4943b1",
         "os-extended-snapshot-
attributes:project_id":"0c2eba2c5af04d3f9e9d0d410b371fde",
         "size":1,
         "id":"e820db06-58b5-439d-bac6-c01faa3f6499",
         "name":"snap-002"
      }
   ]
}
```

This operation does not return a response body.

Show snapshot information

Method	URI	Description
GET	`/v2/{tenant_id}/snapshots/{snapshot_id}`	Shows information for a specified snapshot.

Normal response codes: 200

Request

This table shows the URI parameters for the show snapshot information request:

Name	Type	Description
`{tenant_id}`	String	The unique identifier of the tenant or account.
`{snapshot_id}`	UUID	The unique identifier of an existing snapshot.

This operation does not require a request body.

Response

Example 3.23. Show snapshot information: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<snapshot
  xmlns:os-extended-snapshot-attributes="http://docs.openstack.org/openstack-
block-storage/2.0/content/Extended_Snapshot_Attributes.html"
  status="available" description="Very important"
  created_at="2013-02-25 04:13:17"
  volume_id="5aa119a8-d25b-45a7-8d1b-88e127885635" size="1"
  id="2bb856e1-b3d8-4432-a858-09e4ce939389" name="snap-001"
  os-extended-snapshot-attributes:project_id=
"0c2eba2c5af04d3f9e9d0d410b371fde"
  os-extended-snapshot-attributes:progress="100%">
  <metadata/>
</snapshot>
```

Example 3.24. Show snapshot information: JSON response

```
{
   "snapshot":{
      "status":"available",
      "os-extended-snapshot-attributes:progress":"100%",
      "description":"Daily backup",
      "created_at":"2013-02-25T04:13:17.000000",
      "metadata":{

      },
      "volume_id":"5aa119a8-d25b-45a7-8d1b-88e127885635",
      "os-extended-snapshot-
attributes:project_id":"0c2eba2c5af04d3f9e9d0d410b371fde",
      "size":1,
      "id":"2bb856e1-b3d8-4432-a858-09e4ce939389",
      "name":"snap-001"
   }
}
```

This operation does not return a response body.

Update snapshot

Method	URI	Description
PUT	`/v2/{tenant_id}/snapshots/{snapshot_id}{?display_description,display_name}`	Updates a specified snapshot.

Normal response codes: 200

Request

This table shows the URI parameters for the update snapshot request:

Name	Type	Description
`{tenant_id}`	String	The unique identifier of the tenant or account.
`{snapshot_id}`	UUID	The unique identifier of an existing snapshot.

This table shows the query parameters for the update snapshot request:

Name	Type	Description
`display_description`	String *(Optional)*	Describes the snapshot.
`display_name`	String *(Optional)*	The name of the snapshot.

Example 3.25. Update snapshot: XML request

```
<?xml version="1.0" encoding="UTF-8"?>
<snapshot
    xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
    display_name="snap-002" display_description="This is yet, another
 snapshot."/>
```

Example 3.26. Update snapshot: JSON request

```
{
   "snapshot":{
      "display_name":"snap-002",
      "display_description":"This is yet, another snapshot."
   }
}
```

This operation does not require a request body.

Response

Example 3.27. Update snapshot: XML response

```
<?xml version='1.0' encoding='UTF-8'?>
<snapshot
  xmlns:os-extended-snapshot-attributes="http://docs.openstack.org/openstack-
block-storage/2.0/content/Extended_Snapshot_Attributes.html"
```

```
    status="available"
    display_description="This is yet, another snapshot"
    created_at="2013-02-20T08:11:34.000000"
    volume_id="2402b902-0b7a-458c-9c07-7435a826f794"
    size="1"
    id="4b502fcb-1f26-45f8-9fe5-3b9a0a52eaf2"
    display_name="vol-002"
    os-extended-snapshot-attributes:project_id=
"0c2eba2c5af04d3f9e9d0d410b371fde"
    os-extended-snapshot-attributes:progress="100%">
    <metadata/>
</snapshot>
```

Example 3.28. Update snapshot: JSON response

```
{
   "snapshot":{
      "created_at":"2013-02-20T08:11:34.000000",
      "display_description":"This is yet, another snapshot",
      "display_name":"vol-002",
      "id":"4b502fcb-1f26-45f8-9fe5-3b9a0a52eaf2",
      "size":1,
      "status":"available",
      "volume_id":"2402b902-0b7a-458c-9c07-7435a826f794"
   }
}
```

This operation does not return a response body.

Delete snapshot

Method	URI	Description
DELETE	`/v2/{tenant_id}/snapshots/{snapshot_id}`	Deletes a specified snapshot.

Normal response codes: 202

Request

This table shows the URI parameters for the delete snapshot request:

Name	Type	Description
`{tenant_id}`	String	The unique identifier of the tenant or account.
`{snapshot_id}`	UUID	The unique identifier of an existing snapshot.

This operation does not require a request body.

Volume types

Method	URI	Description
GET	`/v2/{tenant_id}/types`	Lists volume types.
GET	`/v2/{tenant_id}/types/{volume_type_id}`	Shows information about a specified volume type.

List volume types

Method	URI	Description
GET	`/v2/{tenant_id}/types`	Lists volume types.

Normal response codes: 200

Request

This table shows the URI parameters for the list volume types request:

Name	Type	Description
`{tenant_id}`	String	The unique identifier of the tenant or account.

This operation does not require a request body.

Response

Example 3.29. List volume types: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_types
    xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content">
    <volume_type id="6685584b-1eac-4da6-b5c3-555430cf68ff" name="SSD">
        <extra_specs>
            <extra_spec key="capabilities">gpu</extra_spec>
        </extra_specs>
    </volume_type>
    <volume_type id="8eb69a46-df97-4e41-9586-9a40a7533803" name="SATA"
    />
</volume_types>
```

Example 3.30. List volume types: JSON response

```
{
   "volume_types":[
      {
         "extra_specs":{
            "capabilities":"gpu"
         },
         "id":"6685584b-1eac-4da6-b5c3-555430cf68ff",
         "name":"SSD"
      },
      {
         "extra_specs":{

         },
         "id":"8eb69a46-df97-4e41-9586-9a40a7533803",
         "name":"SATA"
      }
   ]
}
```

This operation does not return a response body.

Show volume type information

Method	URI	Description
GET	`/v2/{tenant_id}/types/{volume_type_id}`	Shows information about a specified volume type.

Normal response codes: 200

Request

This table shows the URI parameters for the show volume type information request:

Name	Type	Description
`{tenant_id}`	String	The unique identifier of the tenant or account.
`{volume_type_id}`	UUID	The unique identifier of an existing volume type.

This operation does not require a request body.

Response

Example 3.31. Show volume type information: JSON response

```
{
   "volume_type":{
      "id":"6685584b-1eac-4da6-b5c3-555430cf68ff",
      "name":"SSD",
      "extra_specs":{
         "capabilities":"gpu"
      }
   }
}
```

Example 3.32. Show volume type information: XML response

```
<?xml version="1.0" encoding="UTF-8"?>
<volume_type
    xmlns="http://docs.openstack.org/openstack-block-storage/2.0/content"
    id="6685584b-1eac-4da6-b5c3-555430cf68ff" name="SSD">
    <extra_specs>
        <extra_spec key="capabilities">gpu</extra_spec>
    </extra_specs>
</volume_type>
```

This operation does not return a response body.

Extensions

You can extend the Block Storage API. Extensions are add-ons to the API that enable new features.

Backups

A backup is a full copy of a volume stored in an external service. The service can be configured. The only supported service for now is Object Storage. A backup can subsequently be restored from the external service to either the same volume that the backup was originally taken from, or to a new volume. backup and restore operations can only be carried out on volumes which are in an unattached and available state.

When you create, list, or delete backups, these status values are possible:

Status	Description
creating	The backup is being created.
available	The backup is ready to be restored to a volume.
deleting	The backup is being deleted.
error	An error has occurred with the backup.
restoring	The backup is being restored to a volume.
error_restoring	An error occurred during backup restoration to a volume.

In the event of an error, more information about the error can be found in the FAIL_REASON field for the backup.

Verb	URI	Description
GET	/backups	Lists backups defined in Block Storage that the tenant who submits the request can access.
GET	/backups/detail	Lists detailed information for backups defined in Block Storage that the tenant who submits the request can access.
GET	/backups/*backup_id*	Lists detailed information for a specified backup.
POST	/backups	Creates a Block Storage backup from a volume.
DELETE	/backups/*backup_id*	Deletes a specified backup.
POST	/backups/*backup_id*/restore	Restores a Block Storage backup to an existing or new volume.

Create backup

Verb	URI	Description
POST	/backups	Creates a Block Storage backup from a volume.

Example 3.33. Create backup request: JSON

```
POST /v2/backups
Content-Type: application/json
Accept: application/json
```

```
{
   "backup":{
      "container":null,
      "description":null,
      "name":"backup001",
      "volume_id":"64f5d2fb-d836-4063-b7e2-544d5c1ff607"
   }
}
```

Example 3.34. Create backup response: JSON

```
{
   "backup":{
      "id":"deac8b8c-35c9-4c71-acaa-889c2d5d5c8e",
      "links":[
         {
            "href":"http://localhost:8776/v2/c95fc3e4afe248a49a28828f286a7b38/
backups/deac8b8c-35c9-4c71-acaa-889c2d5d5c8e",
            "rel":"self"
         },
         {
            "href":"http://localhost:8776/c95fc3e4afe248a49a28828f286a7b38/
backups/deac8b8c-35c9-4c71-acaa-889c2d5d5c8e",
            "rel":"bookmark"
         }
      ],
      "name":"backup001"
   }
}
```

Returns status code 202 on success.

List backups

Verb	URI	Description
GET	/backups	Lists backups defined in Block Storage that the tenant who submits the request can access.

Example 3.35. List backups request: JSON

```
GET /v2/backups
Accept: application/json
```

Example 3.36. List backups response: JSON

```
{
   "backups":[
      {
         "id":"2ef47aee-8844-490c-804d-2a8efe561c65",
         "links":[
            {
               "href":"http://localhost:8776/
v1/c95fc3e4afe248a49a28828f286a7b38/backups/
2ef47aee-8844-490c-804d-2a8efe561c65",
               "rel":"self"
            },
            {
               "href":"http://localhost:8776/c95fc3e4afe248a49a28828f286a7b38/
backups/2ef47aee-8844-490c-804d-2a8efe561c65",
               "rel":"bookmark"
            }
         ],
         "name":"backup001"
      },
      {
         "id":"4dbf0ec2-0b57-4669-9823-9f7c76f2b4f8",
         "links":[
            {
               "href":"http://localhost:8776/
v1/c95fc3e4afe248a49a28828f286a7b38/backups/
4dbf0ec2-0b57-4669-9823-9f7c76f2b4f8",
               "rel":"self"
            },
            {
               "href":"http://localhost:8776/c95fc3e4afe248a49a28828f286a7b38/
backups/4dbf0ec2-0b57-4669-9823-9f7c76f2b4f8",
               "rel":"bookmark"
            }
         ],
         "name":"backup002"
      }
   ]
}
```

Returns status code 200 on success.

List backup details

Verb	URI	Description
GET	/backups/details	Lists detailed information for backups defined in Block Storage that the tenant who submits the request can access.

Example 3.37. List backups request: JSON

```
GET /v2/backups/details
Accept: application/json
```

Example 3.38. List backups response: JSON

```
{
   "backups":[
      {
         "availability_zone":"az1",
         "container":"volumebackups",
         "created_at":"2013-04-02T10:35:27.000000",
         "description":null,
         "fail_reason":null,
         "id":"2ef47aee-8844-490c-804d-2a8efe561c65",
         "links":[
            {
               "href":"http://localhost:8776/
v1/c95fc3e4afe248a49a28828f286a7b38/backups/
2ef47aee-8844-490c-804d-2a8efe561c65",
               "rel":"self"
            },
            {
               "href":"http://localhost:8776/c95fc3e4afe248a49a28828f286a7b38/
backups/2ef47aee-8844-490c-804d-2a8efe561c65",
               "rel":"bookmark"
            }
         ],
         "name":"backup001",
         "object_count":22,
         "size":1,
         "status":"available",
         "volume_id":"e5185058-943a-4cb4-96d9-72c184c337d6"
      },
      {
         "availability_zone":"az1",
         "container":"volumebackups",
         "created_at":"2013-04-02T10:21:48.000000",
         "description":null,
         "fail_reason":null,
         "id":"4dbf0ec2-0b57-4669-9823-9f7c76f2b4f8",
         "links":[
            {
               "href":"http://localhost:8776/
v1/c95fc3e4afe248a49a28828f286a7b38/backups/
4dbf0ec2-0b57-4669-9823-9f7c76f2b4f8",
               "rel":"self"
            },
            {
               "href":"http://localhost:8776/c95fc3e4afe248a49a28828f286a7b38/
backups/4dbf0ec2-0b57-4669-9823-9f7c76f2b4f8",
               "rel":"bookmark"
            }
         ],
         "name":"backup002",
         "object_count":22,
         "size":1,
         "status":"available",
         "volume_id":"e5185058-943a-4cb4-96d9-72c184c337d6"
      }
   ]
}
```

Returns status code 200 on success.

Show backup

Verb	URI	Description
GET	/backups/*backup-id*	Lists detailed information for a specified backup ID.

Example 3.39. Show backup request: JSON

```
GET /v2/backups/fa4351c8-d881-4e31-904f-c28a8a2e80cc
Accept: application/json
```

Example 3.40. Show backup response: JSON

```
{
   "backup":{
      "availability_zone":"az1",
      "container":"volumebackups",
      "created_at":"2013-04-02T10:35:27.000000",
      "description":null,
      "fail_reason":null,
      "id":"2ef47aee-8844-490c-804d-2a8efe561c65",
      "links":[
         {
            "href":"http://localhost:8776/v1/c95fc3e4afe248a49a28828f286a7b38/
backups/2ef47aee-8844-490c-804d-2a8efe561c65",
            "rel":"self"
         },
         {
            "href":"http://localhost:8776/c95fc3e4afe248a49a28828f286a7b38/
backups/2ef47aee-8844-490c-804d-2a8efe561c65",
            "rel":"bookmark"
         }
      ],
      "name":"backup001",
      "object_count":22,
      "size":1,
      "status":"available",
      "volume_id":"e5185058-943a-4cb4-96d9-72c184c337d6"
   }
}
```

Returns status code 200 on success.

Delete backup

Verb	URI	Description
DELETE	/backups/*backups-id*	Deletes a specified backup.

Example 3.41. Delete backup: Request

```
DELETE /v2/backups/fa4351c8-d881-4e31-904f-c28a8a2e80cc
```

Example 3.42. Delete backup: Response

```
The response body is empty with status code 202.
```

Restore backup

Verb	URI	Description
POST	/backups/*backup_id*/restore	Restores a Block Storage backup to an existing or new Block Storage volume.

Example 3.43. Restore backup request: JSON

```
POST /v2/backups/fa4351c8-d881-4e31-904f-c28a8a2e80cc/restore
Content-Type: application/json
Accept: application/json
```

```
{
   "restore":{
      "volume_id":"64f5d2fb-d836-4063-b7e2-544d5c1ff607"
   }
}
```

Example 3.44. Restore backup response: JSON

```
{
   "restore":{
      "backup_id":"2ef47aee-8844-490c-804d-2a8efe561c65",
      "volume_id":"795114e8-7489-40be-a978-83797f2c1dd3"
   }
}
```

Returns the 202 status code on success.