Cinder Client Documentation

Release 9.3.1.dev2

Cinder Contributors

Feb 24, 2023

CONTENTS

In order to use the Python api directly, you must first obtain an auth token and identify which endpoint you wish to speak to. Once you have done so, you can use the API like so:

```
>>> from cinderclient import client
>>> cinder = client.Client('1', $OS_USER_NAME, $OS_PASSWORD, $OS_PROJECT_NAME,
↪ $OS_AUTH_URL)
>>> cinder.volumes.list()
[]
>>> myvol = cinder.volumes.create(display_name="test-vol", size=1)
>>> myvol.id
ce06d0a8-5c1b-4e2c-81d2-39eca6bbfb70
>>> cinder.volumes.list()
[<Volume: ce06d0a8-5c1b-4e2c-81d2-39eca6bbfb70>]
>>> myvol.delete()
```

Alternatively, you can create a client instance using the keystoneauth session API:

```
>>> from keystoneauth1 import loading
>>> from keystoneauth1 import session
>>> from cinderclient import client
>>> loader = loading.get_plugin_loader('password')
>>> auth = loader.load_from_options(auth_url=AUTH_URL,
...                                 username=USERNAME,
...                                 password=PASSWORD,
...                                 project_id=PROJECT_ID,
...                                 user_domain_name=USER_DOMAIN_NAME)
>>> sess = session.Session(auth=auth)
>>> cinder = client.Client(VERSION, session=sess)
>>> cinder.volumes.list()
[]
```

USER GUIDES

1.1 The `cinder` shell utility

The **cinder** shell utility interacts with the OpenStack Cinder API from the command line. It supports the entirety of the OpenStack Cinder API.

Youll need to provide **cinder** with your OpenStack username and API key. You can do this with the *os-username*, *os-password* and *os-tenant-name* options, but its easier to just set them as environment variables by setting two environment variables:

OS_USERNAME or CINDER_USERNAME

> Your OpenStack Cinder username.

OS_PASSWORD or CINDER_PASSWORD

> Your password.

OS_PROJECT_NAME or CINDER_PROJECT_ID

> Project for work.

OS_AUTH_URL or CINDER_URL

> The OpenStack API server URL.

OS_VOLUME_API_VERSION

> The OpenStack Block Storage API version.

For example, in Bash youd use:

```
export OS_USERNAME=yourname
export OS_PASSWORD=yadayadayada
export OS_PROJECT_NAME=myproject
export OS_AUTH_URL=http://auth.example.com:5000/v3
export OS_VOLUME_API_VERSION=3
```

If OS_VOLUME_API_VERSION is not set, the highest version supported by the server will be used.

If OS_VOLUME_API_VERSION exceeds the highest version supported by the server, the highest version supported by both the client and server will be used. A warning message is printed when this occurs.

From there, all shell commands take the form:

```
cinder <command> [arguments...]
```

Run **cinder help** to get a full list of all possible commands, and run **cinder help <command>** to get detailed help for that command.

1.2 Using noauth

1.2.1 Cinder Server side API setup

The changes in the cinder.conf on your cinder-api node are minimal, just set authstrategy to noauth:

```
[DEFAULT]
auth_strategy = noauth
...
```

Using cinderclient

To use the cinderclient youll need to set the following env variables:

```
OS_AUTH_TYPE=noauth
CINDER_ENDPOINT=http://<cinder-api-url>:8776/v3
OS_PROJECT_ID=foo
OS_VOLUME_API_VERSION=3.10
```

Note that you can have multiple projects, however we dont currently do any sort of authentication of ownership because, well thats the whole point, its noauth.

Each of these options can also be specified on the cmd line:

```
cinder --os-auth-type=noauth \
--os-endpoint=http://<cinder-api-url>:8776/v3 \
--os-project-id=admin \
--os-volume-api-version=3.10 list
```

COMMAND-LINE REFERENCE

2.1 `cinder` CLI man page

2.1.1 SYNOPSIS

cinder [options] <command> [command-options]

cinder help

cinder help <command>

2.1.2 DESCRIPTION

The **cinder** command line utility interacts with OpenStack Block Storage Service (Cinder).

In order to use the CLI, you must provide your OpenStack username, password, project (historically called tenant), and auth endpoint. You can use configuration options *os-username*, *os-password*, *os-project-name* or *os-project-id*, and *os-auth-url* or set corresponding environment variables:

```
export OS_USERNAME=user
export OS_PASSWORD=pass
export OS_PROJECT_NAME=myproject
export OS_AUTH_URL=http://auth.example.com:5000/v3
```

You can select an API version to use by *os-volume-api-version* option or by setting corresponding environment variable:

```
export OS_VOLUME_API_VERSION=3
```

2.1.3 OPTIONS

To get a list of available commands and options run:

```
cinder help
```

To get usage and options of a command:

```
cinder help <command>
```

You can see more details about the Cinder Command-Line Client at *Block Storage service (cinder) command-line client*.

2.1.4 EXAMPLES

Get information about volume create command:

```
cinder help create
```

List all the volumes:

```
cinder list
```

Create new volume:

```
cinder create 1 --name volume01
```

Describe a specific volume:

```
cinder show 65d23a41-b13f-4345-ab65-918a4b8a6fe6
```

Create a snapshot:

```
cinder snapshot-create 65d23a41-b13f-4345-ab65-918a4b8a6fe6 \
                       --name qt-snap
```

2.1.5 BUGS

Cinder client is hosted in Launchpad so you can view current bugs at https://bugs.launchpad.net/python-cinderclient/.

2.2 Block Storage service (cinder) command-line client

The cinder client is the command-line interface (CLI) for the Block Storage service (cinder) API and its extensions.

For help on a specific **cinder** command, enter:

```
$ cinder help COMMAND
```

2.2.1 cinder usage

```
usage: cinder [--version] [-d] [--service-type <service-type>]
              [--service-name <service-name>]
              [--volume-service-name <volume-service-name>]
              [--os-endpoint-type <os-endpoint-type>]
              [--os-volume-api-version <volume-api-ver>]
              [--os-endpoint <os-endpoint>] [--retries <retries>] [--insecure]
```

(continues on next page)

```
                    [--os-cacert <ca-certificate>] [--os-cert <certificate>]
                    [--os-key <key>] [--timeout <seconds>] [--collect-timing]
                    [--os-auth-type <name>] [--os-auth-url OS_AUTH_URL]
                    [--os-system-scope OS_SYSTEM_SCOPE]
                    [--os-domain-id OS_DOMAIN_ID] [--os-domain-name OS_DOMAIN_NAME]
                    [--os-project-id OS_PROJECT_ID]
                    [--os-project-name OS_PROJECT_NAME]
                    [--os-project-domain-id OS_PROJECT_DOMAIN_ID]
                    [--os-project-domain-name OS_PROJECT_DOMAIN_NAME]
                    [--os-trust-id OS_TRUST_ID]
                    [--os-default-domain-id OS_DEFAULT_DOMAIN_ID]
                    [--os-default-domain-name OS_DEFAULT_DOMAIN_NAME]
                    [--os-user-id OS_USER_ID] [--os-username OS_USERNAME]
                    [--os-user-domain-id OS_USER_DOMAIN_ID]
                    [--os-user-domain-name OS_USER_DOMAIN_NAME]
                    [--os-password OS_PASSWORD] [--os-auth-strategy <auth-strategy>]
                    [--os-token <token>] [--os-url <url>]
                      <subcommand> ...
```

Optional Arguments

--collect-timing Collect per-API call timing information.

--insecure Explicitly allow client to perform insecure TLS (https) requests. The servers certificate will not be verified against any certificate authorities. This option should be used with caution.

--os-auth-strategy Authentication strategy (Env: OS_AUTH_STRATEGY, default keystone). For now, any other value will disable the authentication.

--os-auth-type, --os-auth-plugin Authentication type to use

--os-auth-url Authentication URL

--os-cacert Specify a CA bundle file to use in verifying a TLS (https) server certificate. Defaults to env[OS_CACERT].

--os-cert The location for the keystore (PEM formatted) containing the public key of this client. Defaults to env[OS_CERT].

--os-default-domain-id Optional domain ID to use with v3 and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

--os-default-domain-name Optional domain name to use with v3 API and v2 parameters. It will be used for both the user and project domain in v3 and ignored in v2 authentication.

--os-domain-id Domain ID to scope to

--os-domain-name Domain name to scope to

--os-endpoint Use this API endpoint instead of the Service Catalog. Defaults to env[CINDER_ENDPOINT].

--os-endpoint-type Endpoint type, which is publicURL or internalURL. Default=env[OS_ENDPOINT_TYPE] or nova env[CINDER_ENDPOINT_TYPE] or publicURL.

--os-key The location for the keystore (PEM formatted) containing the private key of this client. Defaults to env[OS_KEY].

--os-password Users password

--os-project-domain-id Domain ID containing project

--os-project-domain-name Domain name containing project

--os-project-id, --os-tenant-id Project ID to scope to

--os-project-name, --os-tenant-name Project name to scope to

--os-system-scope Scope for system operations

--os-token Defaults to env[OS_TOKEN].

--os-trust-id ID of the trust to use as a trustee use

--os-url Defaults to env[OS_URL].

--os-user-domain-id Users domain id

--os-user-domain-name Users domain name

--os-user-id User id

--os-username, --os-user-name Username

--os-volume-api-version Block Storage API version. Accepts X, X.Y (where X is major and Y is minor part). NOTE: this client accepts only 3 for the major version. Default=env[OS_VOLUME_API_VERSION].

--retries Number of retries.

--service-name Service name. Default=env[CINDER_SERVICE_NAME].

--service-type Service type. For most actions, default is volume.

--timeout Set request timeout (in seconds).

--version show programs version number and exit

--volume-service-name Volume service name. Default=env[CINDER_VOLUME_SERVICE_NAME].

-d, --debug Shows debugging output.

Commands

absolute-limits Lists absolute limits for a user.

api-version Display the server API version information.

attachment-complete Complete an attachment for a cinder volume.

attachment-create Create an attachment for a cinder volume.

attachment-delete Delete an attachment for a cinder volume.

attachment-list Lists all attachments.

attachment-show Show detailed information for attachment.

attachment-update Update an attachment for a cinder volume.

availability-zone-list Lists all availability zones.

backup-create Creates a volume backup.

backup-delete Removes one or more backups.

backup-export Export backup metadata record.

backup-import Import backup metadata record.

backup-list Lists all backups.

backup-reset-state Explicitly updates the backup state.

backup-restore Restores a backup.

backup-show Shows backup details.

backup-update Updates a backup.

cgsnapshot-create Creates a cgsnapshot.

cgsnapshot-delete Removes one or more cgsnapshots.

cgsnapshot-list Lists all cgsnapshots.

cgsnapshot-show Shows cgsnapshot details.

cluster-disable Disables clustered services.

cluster-enable Enables clustered services.

cluster-list Lists clustered services with optional filtering.

cluster-show Show detailed information on a clustered service.

consisgroup-create Creates a consistency group.

consisgroup-create-from-src Creates a consistency group from a cgsnapshot or a source CG.

consisgroup-delete Removes one or more consistency groups.

consisgroup-list Lists all consistency groups.

consisgroup-show Shows details of a consistency group.

consisgroup-update Updates a consistency group.

create Creates a volume.

default-type-list Lists all default volume types.

default-type-set Sets a default volume type for a project.

default-type-unset Unset default volume types.

delete Removes one or more volumes.

encryption-type-create Creates encryption type for a volume type. Admin only.

encryption-type-delete Deletes encryption type for a volume type. Admin only.

encryption-type-list Shows encryption type details for volume types. Admin only.

encryption-type-show Shows encryption type details for a volume type. Admin only.

encryption-type-update Update encryption type information for a volume type (Admin Only).

extend Attempts to extend size of an existing volume.

extra-specs-list Lists current volume types and extra specs.

failover-host Failover a replicating cinder-volume host.

force-delete Attempts force-delete of volume, regardless of state.

freeze-host Freeze and disable the specified cinder-volume host.

get-capabilities Show backend volume stats and properties. Admin only.

get-pools Show pool information for backends. Admin only.

group-create Creates a group.

group-create-from-src Creates a group from a group snapshot or a source group.

group-delete Removes one or more groups.

group-disable-replication Disables replication for group.

group-enable-replication Enables replication for group.

group-failover-replication Fails over replication for group.

group-list Lists all groups.

group-list-replication-targets Lists replication targets for group.

group-show Shows details of a group.

group-snapshot-create Creates a group snapshot.

group-snapshot-delete Removes one or more group snapshots.

group-snapshot-list Lists all group snapshots.

group-snapshot-show Shows group snapshot details.

group-specs-list Lists current group types and specs.

group-type-create Creates a group type.

group-type-default List the default group type.

group-type-delete Deletes group type or types.

group-type-key Sets or unsets group_spec for a group type.

group-type-list Lists available group types. (Admin only will see private types)

group-type-show Show group type details.

group-type-update Updates group type name, description, and/or is_public.

group-update Updates a group.

image-metadata Sets or deletes volume image metadata.

image-metadata-show Shows volume image metadata.

list Lists all volumes.

list-filters List enabled filters.

manage Manage an existing volume.

manageable-list Lists all manageable volumes.

message-delete Removes one or more messages.

message-list Lists all messages.

message-show Shows message details.

metadata Sets or deletes volume metadata.

metadata-show Shows volume metadata.

metadata-update-all Updates volume metadata.

migrate Migrates volume to a new host.

qos-associate Associates qos specs with specified volume type.

qos-create Creates a qos specs.

qos-delete Deletes a specified qos specs.

qos-disassociate Disassociates qos specs from specified volume type.

qos-disassociate-all Disassociates qos specs from all its associations.

qos-get-association Lists all associations for specified qos specs.

qos-key Sets or unsets specifications for a qos spec.

qos-list Lists qos specs.

qos-show Shows qos specs details.

quota-class-show Lists quotas for a quota class.

quota-class-update Updates quotas for a quota class.

quota-defaults Lists default quotas for a tenant.

quota-delete Delete the quotas for a tenant.

quota-show Lists quotas for a tenant.

quota-update Updates quotas for a tenant.

quota-usage Lists quota usage for a tenant.

rate-limits Lists rate limits for a user.

readonly-mode-update Updates volume read-only access-mode flag.

reimage Rebuilds a volume, overwriting all content with the specified image

rename Renames a volume.

reset-state Explicitly updates the entity state in the Cinder database.

retype Changes the volume type for a volume.

revert-to-snapshot Revert a volume to the specified snapshot.

service-disable Disables the service.

service-enable Enables the service.

service-get-log Gets the service log level.

2.2. Block Storage service (cinder) command-line client 11

service-list Lists all services. Filter by host and service binary.

service-set-log Sets the service log level.

set-bootable Update bootable status of a volume.

show Shows volume details.

snapshot-create Creates a snapshot.

snapshot-delete Removes one or more snapshots.

snapshot-list Lists all snapshots.

snapshot-manage Manage an existing snapshot.

snapshot-manageable-list Lists all manageable snapshots.

snapshot-metadata Sets or deletes snapshot metadata.

snapshot-metadata-show Shows snapshot metadata.

snapshot-metadata-update-all Updates snapshot metadata.

snapshot-rename Renames a snapshot.

snapshot-reset-state Explicitly updates the snapshot state.

snapshot-show Shows snapshot details.

snapshot-unmanage Stop managing a snapshot.

summary Get volumes summary.

thaw-host Thaw and enable the specified cinder-volume host.

transfer-accept Accepts a volume transfer.

transfer-create Creates a volume transfer.

transfer-delete Undoes a transfer.

transfer-list Lists all transfers.

transfer-show Shows transfer details.

type-access-add Adds volume type access for the given project.

type-access-list Print access information about the given volume type.

type-access-remove Removes volume type access for the given project.

type-create Creates a volume type.

type-default List the default volume type.

type-delete Deletes volume type or types.

type-key Sets or unsets extra_spec for a volume type.

type-list Lists available volume types.

type-show Show volume type details.

type-update Updates volume type name, description, and/or is_public.

unmanage Stop managing a volume.

upload-to-image Uploads volume to Image Service as an image.

version-list List all API versions.

work-cleanup Request cleanup of services with optional filtering.

help Shows help about this program or one of its subcommands.

2.2.2 Command Details

cinder absolute-limits

```
usage: cinder absolute-limits [<tenant_id>]
```

Lists absolute limits for a user.

Positional arguments:

<tenant_id> Display information for a single tenant (Admin only).

cinder api-version

```
usage: cinder api-version
```

Display the server API version information.

cinder attachment-complete

```
usage: cinder attachment-complete <attachment> [<attachment> ...]
```

Complete an attachment for a cinder volume.

Positional arguments:

<attachment> ID of attachment or attachments to delete.

cinder attachment-create

```
usage: cinder attachment-create [--connect <connect>]
                                [--initiator <initiator>] [--ip <ip>]
                                [--host <host>] [--platform <platform>]
                                [--ostype <ostype>] [--multipath <multipath>]
                                [--mountpoint <mountpoint>] [--mode <mode>]
                                <volume> [<server_id>]
```

Create an attachment for a cinder volume.

Positional arguments:

<volume> Name or ID of volume or volumes to attach.

<server_id> ID of server attaching to.

Optional arguments:

--connect <connect> Make an active connection using provided connector info (True or False).

--initiator <initiator> iqn of the initiator attaching to. Default=None.

--ip <ip> ip of the system attaching to. Default=None.

--host <host> Name of the host attaching to. Default=None.

--platform <platform> Platform type. Default=x86_64.

--ostype <ostype> OS type. Default=linux2.

--multipath <multipath> Use multipath. Default=False.

--mountpoint <mountpoint> Mountpoint volume will be attached at. Default=None.

--mode <mode> Mode of attachment, rw, ro and null, where null indicates we want to honor any existing admin-metadata settings. Default=null.

cinder attachment-delete

```
usage: cinder attachment-delete <attachment> [<attachment> ...]
```

Delete an attachment for a cinder volume.

Positional arguments:

<attachment> ID of attachment or attachments to delete.

cinder attachment-list

```
usage: cinder attachment-list [--all-tenants [<0|1>]]
                              [--volume-id <volume-id>] [--status <status>]
                              [--marker <marker>] [--limit <limit>]
                              [--sort <key>[:<direction>]]
                              [--tenant [<tenant>]]
                              [--filters [<key=value> ...]]
```

Lists all attachments.

Optional arguments:

--all-tenants <0|1> Shows details for all tenants. Admin only.

--volume-id <volume-id> Filters results by a volume ID. Default=None. This option is deprecated and will be removed in newer release. Please use filters option which is introduced since 3.33 instead.

--status <status> Filters results by a status. Default=None. This option is deprecated and will be removed in newer release. Please use filters option which is introduced since 3.33 instead.

--marker <marker> Begin returning attachments that appear later in attachment list than that represented by this id. Default=None.

--limit <limit> Maximum number of attachments to return. Default=None.

--sort <key>[:<direction>] Comma-separated list of sort keys and directions in the form of <key>[:<asc|desc>]. Valid keys: id, status, size, availability_zone, name, bootable, created_at, reference. Default=None.

--tenant <tenant> Display information from single tenant (Admin only).

--filters <key=value> Filter key and value pairs. Please use cinder list-filters to check enabled filters from server. Use key~=value for inexact filtering if the key supports. Default=None.

cinder attachment-show

```
usage: cinder attachment-show <attachment>
```

Show detailed information for attachment.

Positional arguments:

<attachment> ID of attachment.

cinder attachment-update

```
usage: cinder attachment-update [--initiator <initiator>] [--ip <ip>]
                                [--host <host>] [--platform <platform>]
                                [--ostype <ostype>] [--multipath <multipath>]
                                [--mountpoint <mountpoint>]
                                <attachment>
```

Update an attachment for a cinder volume.

Positional arguments:

<attachment> ID of attachment.

Optional arguments:

--initiator <initiator> iqn of the initiator attaching to. Default=None.

--ip <ip> ip of the system attaching to. Default=None.

--host <host> Name of the host attaching to. Default=None.

--platform <platform> Platform type. Default=x86_64.

--ostype <ostype> OS type. Default=linux2.

--multipath <multipath> Use multipath. Default=False.

--mountpoint <mountpoint> Mountpoint volume will be attached at. Default=None.

cinder availability-zone-list

```
usage: cinder availability-zone-list
```

Lists all availability zones.

cinder backup-create

```
usage: cinder backup-create [--container <container>] [--name <name>]
                            [--description <description>] [--incremental]
                            [--force] [--snapshot-id <snapshot-id>]
                            [--metadata [<key=value> ...]]
                            [--availability-zone AVAILABILITY_ZONE]
                            <volume>
```

Creates a volume backup.

Positional arguments:

<volume> Name or ID of volume to backup.

Optional arguments:

--container <container> Backup container name. Default=None.

--name <name> Backup name. Default=None.

--description <description> Backup description. Default=None.

--incremental None Incremental backup. Default=False.

--force None Allows or disallows backup of a volume when the volume is attached to an instance. If set to True, backs up the volume whether its status is available or in-use. The backup of an in-use volume means your data is crash consistent. Default=False.

--snapshot-id <snapshot-id> ID of snapshot to backup. Default=None.

--metadata <key=value> Metadata key and value pairs. Default=None.

--availability-zone None AZ where the backup should be stored, by default it will be the same as the source.

cinder backup-delete

```
usage: cinder backup-delete [--force] <backup> [<backup> ...]
```

Removes one or more backups.

Positional arguments:

<backup> Name or ID of backup(s) to delete.

Optional arguments:

--force None Allows deleting backup of a volume when its status is other than available or error. Default=False.

---

cinder backup-export

```
usage: cinder backup-export <backup>
```

Export backup metadata record.

Positional arguments:

<backup> ID of the backup to export.

cinder backup-import

```
usage: cinder backup-import <backup_service> <backup_url>
```

Import backup metadata record.

Positional arguments:

<backup_service> Backup service to use for importing the backup.

<backup_url> Backup URL for importing the backup metadata.

cinder backup-list

```
usage: cinder backup-list [--all-tenants [<all_tenants>]] [--name <name>]
                          [--status <status>] [--volume-id <volume-id>]
                          [--marker <marker>] [--limit <limit>]
                          [--sort <key>[:<direction>]]
                          [--filters [<key=value> ...]]
                          [--with-count [<True|False>]]
```

Lists all backups.

Optional arguments:

--all-tenants <all_tenants> Shows details for all tenants. Admin only.

--name <name> Filters results by a name. Default=None. This option is deprecated and will be removed in newer release. Please use filters option which is introduced since 3.33 instead.

--status <status> Filters results by a status. Default=None. This option is deprecated and will be removed in newer release. Please use filters option which is introduced since 3.33 instead.

--volume-id <volume-id> Filters results by a volume ID. Default=None. This option is deprecated and will be removed in newer release. Please use filters option which is introduced since 3.33 instead.

--marker <marker> Begin returning backups that appear later in the backup list than that represented by this id. Default=None.

--limit <limit> Maximum number of backups to return. Default=None.

--sort <key>[:<direction>] Comma-separated list of sort keys and directions in the form of <key>[:<asc|desc>]. Valid keys: id, status, size, availability_zone, name, bootable, created_at, reference. Default=None.

--filters <key=value> Filter key and value pairs. Please use cinder list-filters to check enabled filters from server. Use key~=value for inexact filtering if the key supports. Default=None.

--with-count <True|False> Show total number of backup entities. This is useful when pagination is applied in the request.

cinder backup-reset-state

```
usage: cinder backup-reset-state [--state <state>] <backup> [<backup> ...]
```

Explicitly updates the backup state.

Positional arguments:

<backup> Name or ID of the backup to modify.

Optional arguments:

--state <state> The state to assign to the backup. Valid values are available, error. Default=available.

cinder backup-restore

```
usage: cinder backup-restore [--volume <volume>] [--name <name>]
                             [--volume-type <volume-type>]
                             [--availability-zone <AZ>]
                             <backup>
```

Restores a backup.

Positional arguments:

<backup> Name or ID of backup to restore.

Optional arguments:

--volume <volume> Name or ID of existing volume to which to restore. This is mutually exclusive with name and takes priority. Default=None.

--name <name> Use the name for new volume creation to restore. This is mutually exclusive with volume and volume takes priority. Default=None.

--volume-type <volume-type> Volume type for the new volume creation to restore. This option is not valid when used with the volume option. Default=None.

--availability-zone <AZ> AZ for the new volume creation to restore. By default it will be the same as backup AZ. This option is not valid when used with the volume option. Default=None.

cinder backup-show

```
usage: cinder backup-show <backup>
```

Shows backup details.

Positional arguments:

<backup> Name or ID of backup.

cinder backup-update

```
usage: cinder backup-update [--name [<name>]] [--description <description>]
                            [--metadata [<key=value> ...]]
                            <backup>
```

Updates a backup.

Positional arguments:

<backup> Name or ID of backup to rename.

Optional arguments:

--name <name> New name for backup.

--description <description> Backup description. Default=None.

--metadata <key=value> Metadata key and value pairs. Default=None.

cinder cgsnapshot-create

```
usage: cinder cgsnapshot-create [--name <name>] [--description <description>]
                                <consistencygroup>
```

Creates a cgsnapshot.

Positional arguments:

<consistencygroup> Name or ID of a consistency group.

Optional arguments:

--name <name> Cgsnapshot name. Default=None.

--description <description> Cgsnapshot description. Default=None.

cinder cgsnapshot-delete

```
usage: cinder cgsnapshot-delete <cgsnapshot> [<cgsnapshot> ...]
```

Removes one or more cgsnapshots.

Positional arguments:

<cgsnapshot> Name or ID of one or more cgsnapshots to be deleted.

cinder cgsnapshot-list

```
usage: cinder cgsnapshot-list [--all-tenants [<0|1>]] [--status <status>]
                              [--consistencygroup-id <consistencygroup_id>]
```

Lists all cgsnapshots.

Optional arguments:

--all-tenants <0|1> Shows details for all tenants. Admin only.

--status <status> Filters results by a status. Default=None.

--consistencygroup-id <consistencygroup_id> Filters results by a consistency group ID. Default=None.

cinder cgsnapshot-show

```
usage: cinder cgsnapshot-show <cgsnapshot>
```

Shows cgsnapshot details.

Positional arguments:

<cgsnapshot> Name or ID of cgsnapshot.

cinder cluster-disable

```
usage: cinder cluster-disable [--reason <reason>] [<binary>] <cluster-name>
```

Disables clustered services.

Positional arguments:

<binary> Binary to filter by. Default: cinder-volume.

<cluster-name> Name of the clustered services to update.

Optional arguments:

--reason <reason> Reason for disabling clustered service.

cinder cluster-enable

```
usage: cinder cluster-enable [<binary>] <cluster-name>
```

Enables clustered services.

Positional arguments:

<binary> Binary to filter by. Default: cinder-volume.

<cluster-name> Name of the clustered services to update.

cinder cluster-list

```
usage: cinder cluster-list [--name <name>] [--binary <binary>]
                           [--is-up <True|true|False|false>]
                           [--disabled <True|true|False|false>]
                           [--num-hosts <num-hosts>]
                           [--num-down-hosts <num-down-hosts>] [--detailed]
```

Lists clustered services with optional filtering.

Optional arguments:

--name <name> Filter by cluster name, without backend will list all clustered services from the same cluster. Default=None.

--binary <binary> Cluster binary. Default=None.

--is-up <True|true|False|false> Filter by up/down status. Default=None.

--disabled <True|true|False|false> Filter by disabled status. Default=None.

--num-hosts <num-hosts> Filter by number of hosts in the cluster.

--num-down-hosts <num-down-hosts> Filter by number of hosts that are down.

--detailed None Get detailed clustered service information (Default=False).

cinder cluster-show

```
usage: cinder cluster-show [<binary>] <cluster-name>
```

Show detailed information on a clustered service.

Positional arguments:

<binary> Binary to filter by. Default: cinder-volume.

<cluster-name> Name of the clustered service to show.

cinder consisgroup-create

```
usage: cinder consisgroup-create [--name <name>] [--description <description>]
                                 [--availability-zone <availability-zone>]
                                 <volume-types>
```

Creates a consistency group.

Positional arguments:

<volume-types> Volume types.

Optional arguments:

--name <name> Name of a consistency group.

--description <description> Description of a consistency group. Default=None.

--availability-zone <availability-zone> Availability zone for volume. Default=None.

cinder consisgroup-create-from-src

```
usage: cinder consisgroup-create-from-src [--cgsnapshot <cgsnapshot>]
                                          [--source-cg <source-cg>]
                                          [--name <name>]
                                          [--description <description>]
```

Creates a consistency group from a cgsnapshot or a source CG.

Optional arguments:

--cgsnapshot <cgsnapshot> Name or ID of a cgsnapshot. Default=None.

--source-cg <source-cg> Name or ID of a source CG. Default=None.

--name <name> Name of a consistency group. Default=None.

--description <description> Description of a consistency group. Default=None.

cinder consisgroup-delete

```
usage: cinder consisgroup-delete [--force]
                                 <consistencygroup> [<consistencygroup> ...]
```

Removes one or more consistency groups.

Positional arguments:

<consistencygroup> Name or ID of one or more consistency groups to be deleted.

Optional arguments:

--force None Allows or disallows consistency groups to be deleted. If the consistency group is empty, it can be deleted without the force flag. If the consistency group is not empty, the force flag is required for it to be deleted.

cinder consisgroup-list

```
usage: cinder consisgroup-list [--all-tenants [<0|1>]]
```

Lists all consistency groups.

Optional arguments:

--all-tenants <0|1> Shows details for all tenants. Admin only.

cinder consisgroup-show

```
usage: cinder consisgroup-show <consistencygroup>
```

Shows details of a consistency group.

Positional arguments:

<consistencygroup> Name or ID of a consistency group.

cinder consisgroup-update

```
usage: cinder consisgroup-update [--name <name>] [--description <description>]
                                 [--add-volumes <uuid1,uuid2,......>]
                                 [--remove-volumes <uuid3,uuid4,......>]
                                 <consistencygroup>
```

Updates a consistency group.

Positional arguments:

<consistencygroup> Name or ID of a consistency group.

Optional arguments:

--name <name> New name for consistency group. Default=None.

--description <description> New description for consistency group. Default=None.

--add-volumes <uuid1,uuid2,......> UUID of one or more volumes to be added to the consistency group, separated by commas. Default=None.

--remove-volumes <uuid3,uuid4,......> UUID of one or more volumes to be removed from the consistency group, separated by commas. Default=None.

cinder create

```
usage: cinder create [--consisgroup-id <consistencygroup-id>]
                     [--group-id <group-id>] [--snapshot-id <snapshot-id>]
                     [--source-volid <source-volid>] [--image-id <image-id>]
                     [--image <image>] [--backup-id <backup-id>]
                     [--name <name>] [--description <description>]
                     [--volume-type <volume-type>]
                     [--availability-zone <availability-zone>]
                     [--metadata [<key=value> ...]] [--hint <key=value>]
                     [--poll]
                     [<size>]
```

Creates a volume.

Positional arguments:

`<size>` Size of volume, in GiBs. (Required unless snapshot-id/source-volid/backup-id is specified).

Optional arguments:

`--consisgroup-id <consistencygroup-id>` ID of a consistency group where the new volume belongs to. Default=None.

`--group-id <group-id>` ID of a group where the new volume belongs to. Default=None.

`--snapshot-id <snapshot-id>` Creates volume from snapshot ID. Default=None.

`--source-volid <source-volid>` Creates volume from volume ID. Default=None.

`--image-id <image-id>` Creates volume from image ID. Default=None.

`--image <image>` Creates a volume from image (ID or name). Default=None.

`--backup-id <backup-id>` Creates a volume from backup ID. Default=None.

`--name <name>` Volume name. Default=None.

`--description <description>` Volume description. Default=None.

`--volume-type <volume-type>` Volume type. Default=None, that is, use the default volume type configured for the Block Storage API. You can see what type this is by using the cinder type-default command.

`--availability-zone <availability-zone>` Availability zone for volume. Default=None.

`--metadata <key=value>` Metadata key and value pairs. Default=None.

`--hint <key=value>` Scheduler hint, similar to nova. Repeat option to set multiple hints. Values with the same key will be stored as a list.

`--poll None` Wait for volume creation until it completes.

cinder default-type-list

```
usage: cinder default-type-list [--project-id <project_id>]
```

Lists all default volume types.

Optional arguments:

--project-id <project_id> ID of project for which to show the default type.

cinder default-type-set

```
usage: cinder default-type-set <volume_type> <project_id>
```

Sets a default volume type for a project.

Positional arguments:

<volume_type> Name or ID of the volume type.

<project_id> ID of project for which to set default type.

cinder default-type-unset

```
usage: cinder default-type-unset <project_id> [<project_id> ...]
```

Unset default volume types.

Positional arguments:

<project_id> ID of project for which to unset default type.

cinder delete

```
usage: cinder delete [--cascade] <volume> [<volume> ...]
```

Removes one or more volumes.

Positional arguments:

<volume> Name or ID of volume or volumes to delete.

Optional arguments:

--cascade None Remove any snapshots along with volume. Default=False.

cinder encryption-type-create

```
usage: cinder encryption-type-create [--cipher <cipher>]
                                     [--key-size <key_size>]
                                     [--control-location <control_location>]
                                     <volume_type> <provider>
```

Creates encryption type for a volume type. Admin only.

Positional arguments:

<volume_type> Name or ID of volume type.

<provider> The encryption provider format. For example, luks or plain.

Optional arguments:

--cipher <cipher> The encryption algorithm or mode. For example, aes-xts-plain64. Default=None.

--key-size <key_size> Size of encryption key, in bits. For example, 128 or 256. Default=None.

--control-location <control_location> Notional service where encryption is performed. Valid values are front-end or back-end. For example, front-end=Nova. Default is front-end.

cinder encryption-type-delete

```
usage: cinder encryption-type-delete <volume_type>
```

Deletes encryption type for a volume type. Admin only.

Positional arguments:

<volume_type> Name or ID of volume type.

cinder encryption-type-list

```
usage: cinder encryption-type-list
```

Shows encryption type details for volume types. Admin only.

cinder encryption-type-show

```
usage: cinder encryption-type-show <volume_type>
```

Shows encryption type details for a volume type. Admin only.

Positional arguments:

<volume_type> Name or ID of volume type.

cinder encryption-type-update

```
usage: cinder encryption-type-update [--provider <provider>]
                                     [--cipher [<cipher>]]
                                     [--key-size [<key-size>]]
                                     [--control-location <control-location>]
                                     <volume-type>
```

Update encryption type information for a volume type (Admin Only).

Positional arguments:

<volume-type> Name or ID of the volume type

Optional arguments:

--provider <provider> Encryption provider format (e.g. luks or plain).

--cipher <cipher> Encryption algorithm/mode to use (e.g., aes-xts-plain64). Provide parameter without value to set to provider default.

--key-size <key-size> Size of the encryption key, in bits (e.g., 128, 256). Provide parameter without value to set to provider default.

--control-location <control-location> Notional service where encryption is performed (e.g., front-end=Nova). Values: front-end, back-end

cinder extend

```
usage: cinder extend <volume> <new_size>
```

Attempts to extend size of an existing volume.

Positional arguments:

<volume> Name or ID of volume to extend.

<new_size> New size of volume, in GiBs.

cinder extra-specs-list

```
usage: cinder extra-specs-list
```

Lists current volume types and extra specs.

cinder failover-host

```
usage: cinder failover-host [--backend_id <backend-id>] <hostname>
```

Failover a replicating cinder-volume host.

Positional arguments:

<hostname> Host name.

Optional arguments:

--backend_id <backend-id> ID of backend to failover to (Default=None)

cinder force-delete

```
usage: cinder force-delete <volume> [<volume> ...]
```

Attempts force-delete of volume, regardless of state.

Positional arguments:

<volume> Name or ID of volume or volumes to delete.

cinder freeze-host

```
usage: cinder freeze-host <hostname>
```

Freeze and disable the specified cinder-volume host.

Positional arguments:

<hostname> Host name.

cinder get-capabilities

```
usage: cinder get-capabilities <host>
```

Show backend volume stats and properties. Admin only.

Positional arguments:

<host> Cinder host to show backend volume stats and properties; takes the form: host@backend-name

cinder get-pools

```
usage: cinder get-pools [--detail] [--filters [<key=value> ...]]
```

Show pool information for backends. Admin only.

Optional arguments:

--detail None Show detailed information about pools.

--filters <key=value> Filter key and value pairs. Please use cinder list-filters to check enabled filters from server, Default=None.

cinder group-create

```
usage: cinder group-create [--name <name>] [--description <description>]
                           [--availability-zone <availability-zone>]
                           <group-type> <volume-types>
```

Creates a group.

Positional arguments:

<group-type> Group type.

<volume-types> Comma-separated list of volume types.

Optional arguments:

--name <name> Name of a group.

--description <description> Description of a group. Default=None.

--availability-zone <availability-zone> Availability zone for group. Default=None.

cinder group-create-from-src

```
usage: cinder group-create-from-src [--group-snapshot <group-snapshot>]
                                    [--source-group <source-group>]
                                    [--name <name>]
                                    [--description <description>]
```

Creates a group from a group snapshot or a source group.

Optional arguments:

--group-snapshot <group-snapshot> Name or ID of a group snapshot. Default=None.

--source-group <source-group> Name or ID of a source group. Default=None.

--name <name> Name of a group. Default=None.

--description <description> Description of a group. Default=None.

cinder group-delete

```
usage: cinder group-delete [--delete-volumes] <group> [<group> ...]
```

Removes one or more groups.

Positional arguments:

`<group>` Name or ID of one or more groups to be deleted.

Optional arguments:

`--delete-volumes None` Allows or disallows groups to be deleted if they are not empty. If the group is empty, it can be deleted without the delete-volumes flag. If the group is not empty, the delete-volumes flag is required for it to be deleted. If True, all volumes in the group will also be deleted.

cinder group-disable-replication

```
usage: cinder group-disable-replication <group>
```

Disables replication for group.

Positional arguments:

`<group>` Name or ID of the group.

cinder group-enable-replication

```
usage: cinder group-enable-replication <group>
```

Enables replication for group.

Positional arguments:

`<group>` Name or ID of the group.

cinder group-failover-replication

```
usage: cinder group-failover-replication [--allow-attached-volume]
                                         [--secondary-backend-id <secondary_
↪backend_id>]
                                         <group>
```

Fails over replication for group.

Positional arguments:

`<group>` Name or ID of the group.

Optional arguments:

`--allow-attached-volume None` Allows or disallows group with attached volumes to be failed over.

`--secondary-backend-id <secondary_backend_id>` Secondary backend id. Default=None.

---

cinder group-list

```
usage: cinder group-list [--all-tenants [<0|1>]] [--filters [<key=value> ...]]
```

Lists all groups.

Optional arguments:

`--all-tenants <0|1>` Shows details for all tenants. Admin only.

`--filters <key=value>` Filter key and value pairs. Please use cinder list-filters to check enabled filters from server. Use key~=value for inexact filtering if the key supports. Default=None.

cinder group-list-replication-targets

```
usage: cinder group-list-replication-targets <group>
```

Lists replication targets for group.

Positional arguments:

`<group>` Name or ID of the group.

cinder group-show

```
usage: cinder group-show [--list-volume [<False|True>]] <group>
```

Shows details of a group.

Positional arguments:

`<group>` Name or ID of a group.

Optional arguments:

`--list-volume <False|True>` Shows volumes included in the group.

cinder group-snapshot-create

```
usage: cinder group-snapshot-create [--name <name>]
                                    [--description <description>]
                                    <group>
```

Creates a group snapshot.

Positional arguments:

`<group>` Name or ID of a group.

Optional arguments:

`--name <name>` Group snapshot name. Default=None.

`--description <description>` Group snapshot description. Default=None.

---

cinder group-snapshot-delete

```
usage: cinder group-snapshot-delete <group_snapshot> [<group_snapshot> ...]
```

Removes one or more group snapshots.

Positional arguments:

<group_snapshot> Name or ID of one or more group snapshots to be deleted.

cinder group-snapshot-list

```
usage: cinder group-snapshot-list [--all-tenants [<0|1>]] [--status <status>]
                                  [--group-id <group_id>]
                                  [--filters [<key=value> ...]]
```

Lists all group snapshots.

Optional arguments:

--all-tenants <0|1> Shows details for all tenants. Admin only.

--status <status> Filters results by a status. Default=None. This option is deprecated and will be removed in newer release. Please use filters option which is introduced since 3.33 instead.

--group-id <group_id> Filters results by a group ID. Default=None. This option is deprecated and will be removed in newer release. Please use filters option which is introduced since 3.33 instead.

--filters <key=value> Filter key and value pairs. Please use cinder list-filters to check enabled filters from server. Use key~=value for inexact filtering if the key supports. Default=None.

cinder group-snapshot-show

```
usage: cinder group-snapshot-show <group_snapshot>
```

Shows group snapshot details.

Positional arguments:

<group_snapshot> Name or ID of group snapshot.

cinder group-specs-list

```
usage: cinder group-specs-list
```

Lists current group types and specs.

cinder group-type-create

```
usage: cinder group-type-create [--description <description>]
                                [--is-public <is-public>]
                                <name>
```

Creates a group type.

Positional arguments:

<name> Name of new group type.

Optional arguments:

--description <description> Description of new group type.

--is-public <is-public> Make type accessible to the public (default true).

cinder group-type-default

```
usage: cinder group-type-default
```

List the default group type.

cinder group-type-delete

```
usage: cinder group-type-delete <group_type> [<group_type> ...]
```

Deletes group type or types.

Positional arguments:

<group_type> Name or ID of group type or types to delete.

cinder group-type-key

```
usage: cinder group-type-key <gtype> <action> <key=value> [<key=value> ...]
```

Sets or unsets group_spec for a group type.

Positional arguments:

<gtype> Name or ID of group type.

<action> The action. Valid values are set or unset.

<key=value> The group specs key and value pair to set or unset. For unset, specify only the key.

cinder group-type-list

```
usage: cinder group-type-list [--filters [<key=value> ...]]
```

Lists available group types. (Admin only will see private types)

Optional arguments:

--filters <key=value> Filter key and value pairs. Admin only.

cinder group-type-show

```
usage: cinder group-type-show <group_type>
```

Show group type details.

Positional arguments:

<group_type> Name or ID of the group type.

cinder group-type-update

```
usage: cinder group-type-update [--name <name>] [--description <description>]
                                [--is-public <is-public>]
                                <id>
```

Updates group type name, description, and/or is_public.

Positional arguments:

<id> ID of the group type.

Optional arguments:

--name <name> Name of the group type.

--description <description> Description of the group type.

--is-public <is-public> Make type accessible to the public or not.

cinder group-update

```
usage: cinder group-update [--name <name>] [--description <description>]
                           [--add-volumes <uuid1,uuid2,......>]
                           [--remove-volumes <uuid3,uuid4,......>]
                           <group>
```

Updates a group.

Positional arguments:

<group> Name or ID of a group.

Optional arguments:

--name <name> New name for group. Default=None.

--description <description> New description for group. Default=None.

--add-volumes <uuid1,uuid2,......> UUID of one or more volumes to be added to the group, separated by commas. Default=None.

--remove-volumes <uuid3,uuid4,......> UUID of one or more volumes to be removed from the group, separated by commas. Default=None.

cinder image-metadata

```
usage: cinder image-metadata <volume> <action> <key=value> [<key=value> ...]
```

Sets or deletes volume image metadata.

Positional arguments:

<volume> Name or ID of volume for which to update metadata.

<action> The action. Valid values are set or unset.

<key=value> Metadata key and value pair to set or unset. For unset, specify only the key.

cinder image-metadata-show

```
usage: cinder image-metadata-show <volume>
```

Shows volume image metadata.

Positional arguments:

<volume> ID of volume.

cinder list

```
usage: cinder list [--group_id <group_id>] [--all-tenants [<0|1>]]
                   [--name <name>] [--status <status>]
                   [--bootable [<True|true|False|false>]]
                   [--migration_status <migration_status>]
                   [--metadata [<key=value> ...]]
                   [--image_metadata [<key=value> ...]] [--marker <marker>]
                   [--limit <limit>] [--fields <fields>]
                   [--sort <key>[:<direction>]] [--tenant [<tenant>]]
                   [--filters [<key=value> ...]] [--with-count [<True|False>]]
```

Lists all volumes.

Optional arguments:

--group_id <group_id> Filters results by a group_id. Default=None.This option is deprecated and will be removed in newer release. Please use filters option which is introduced since 3.33 instead.

--all-tenants <0|1> Shows details for all tenants. Admin only.

--name <name> Filters results by a name. Default=None. This option is deprecated and will be removed in newer release. Please use filters option which is introduced since 3.33 instead.

--status <status> Filters results by a status. Default=None. This option is deprecated and will be removed in newer release. Please use filters option which is introduced since 3.33 instead.

--bootable <True|true|False|false> Filters results by bootable status. Default=None. This option is deprecated and will be removed in newer release. Please use filters option which is introduced since 3.33 instead.

--migration_status <migration_status> Filters results by a migration status. Default=None. Admin only. This option is deprecated and will be removed in newer release. Please use filters option which is introduced since 3.33 instead.

--metadata <key=value> Filters results by a metadata key and value pair. Default=None. This option is deprecated and will be removed in newer release. Please use filters option which is introduced since 3.33 instead.

--image_metadata <key=value> Filters results by a image metadata key and value pair. Require volume api version >=3.4. Default=None.This option is deprecated and will be removed in newer release. Please use filters option which is introduced since 3.33 instead.

--marker <marker> Begin returning volumes that appear later in the volume list than that represented by this volume id. Default=None.

--limit <limit> Maximum number of volumes to return. Default=None.

--fields <fields> Comma-separated list of fields to display. Use the show command to see which fields are available. Unavailable/non-existent fields will be ignored. Default=None.

--sort <key>[:<direction>] Comma-separated list of sort keys and directions in the form of <key>[:<asc|desc>]. Valid keys: id, status, size, availability_zone, name, bootable, created_at, reference. Default=None.

--tenant <tenant> Display information from single tenant (Admin only).

--filters <key=value> Filter key and value pairs. Please use cinder list-filters to check enabled filters from server. Use key~=value for inexact filtering if the key supports. Default=None.

--with-count <True|False> Show total number of volume entities. This is useful when pagination is applied in the request.

cinder list-filters

```
usage: cinder list-filters [--resource <resource>]
```

List enabled filters.

Optional arguments:

--resource <resource> Show enabled filters for specified resource. Default=None.

cinder manage

```
usage: cinder manage [--cluster CLUSTER] [--id-type <id-type>] [--name <name>]
                     [--description <description>]
                     [--volume-type <volume-type>]
                     [--availability-zone <availability-zone>]
                     [--metadata [<key=value> ...]] [--bootable]
                     <host> <identifier>
```

Manage an existing volume.

Positional arguments:

`<host>` Cinder host on which the existing volume resides; takes the form: host@backend-name#pool

`<identifier>` Name or other Identifier for existing volume

Optional arguments:

`--cluster None` Cinder cluster on which the existing volume resides; takes the form: cluster@backend-name#pool

`--id-type <id-type>` Type of backend device identifier provided, typically source-name or source-id (Default=source-name)

`--name <name>` Volume name (Default=None)

`--description <description>` Volume description (Default=None)

`--volume-type <volume-type>` Volume type (Default=None)

`--availability-zone <availability-zone>` Availability zone for volume (Default=None)

`--metadata <key=value>` Metadata key=value pairs (Default=None)

`--bootable None` Specifies that the newly created volume should be marked as bootable

cinder manageable-list

```
usage: cinder manageable-list [--detailed <detailed>] [--marker <marker>]
                              [--limit <limit>] [--offset <offset>]
                              [--sort <key>[:<direction>]] [--cluster CLUSTER]
                              [<host>]
```

Lists all manageable volumes.

Positional arguments:

`<host>` Cinder host on which to list manageable volumes; takes the form: host@backend-name#pool

Optional arguments:

`--detailed <detailed>` Returned detailed information (default true).

`--marker <marker>` Begin returning volumes that appear later in the volume list than that represented by this reference. This reference should be json like. Default=None.

`--limit <limit>` Maximum number of volumes to return. Default=None.

`--offset <offset>` Number of volumes to skip after marker. Default=None.

--sort <key>[:<direction>] Comma-separated list of sort keys and directions in the form of
    <key>[:<asc|desc>]. Valid keys: size, reference. Default=None.

--cluster CLUSTER Cinder cluster on which to list manageable volumes; takes the form:
    cluster@backend-name#pool

cinder message-delete

```
usage: cinder message-delete <message> [<message> ...]
```

Removes one or more messages.

Positional arguments:

<message> ID of one or more message to be deleted.

cinder message-list

```
usage: cinder message-list [--marker <marker>] [--limit <limit>]
                           [--sort <key>[:<direction>]]
                           [--resource_uuid <resource_uuid>]
                           [--resource_type <type>] [--event_id <id>]
                           [--request_id <request_id>] [--level <level>]
                           [--filters [<key=value> ...]]
```

Lists all messages.

Optional arguments:

--marker <marker> Begin returning message that appear later in the message list than that represented
    by this id. Default=None.

--limit <limit> Maximum number of messages to return. Default=None.

--sort <key>[:<direction>] Comma-separated list of sort keys and directions in the form of
    <key>[:<asc|desc>]. Valid keys: id, status, size, availability_zone, name, bootable, created_at,
    reference. Default=None.

--resource_uuid <resource_uuid> Filters results by a resource uuid. Default=None. This option
    is deprecated and will be removed in newer release. Please use filters option which is introduced
    since 3.33 instead.

--resource_type <type> Filters results by a resource type. Default=None. This option is deprecated
    and will be removed in newer release. Please use filters option which is introduced since 3.33
    instead.

--event_id <id> Filters results by event id. Default=None. This option is deprecated and will be
    removed in newer release. Please use filters option which is introduced since 3.33 instead.

--request_id <request_id> Filters results by request id. Default=None. This option is deprecated
    and will be removed in newer release. Please use filters option which is introduced since 3.33
    instead.

--level <level> Filters results by the message level. Default=None. This option is deprecated and
    will be removed in newer release. Please use filters option which is introduced since 3.33 instead.

--filters <key=value> Filter key and value pairs. Please use cinder list-filters to check enabled filters from server. Use key~=value for inexact filtering if the key supports. Default=None.

cinder message-show

```
usage: cinder message-show <message>
```

Shows message details.

Positional arguments:

<message> ID of message.

cinder metadata

```
usage: cinder metadata <volume> <action> <key=value> [<key=value> ...]
```

Sets or deletes volume metadata.

Positional arguments:

<volume> Name or ID of volume for which to update metadata.

<action> The action. Valid values are set or unset.

<key=value> Metadata key and value pair to set or unset. For unset, specify only the key(s): <key key>

cinder metadata-show

```
usage: cinder metadata-show <volume>
```

Shows volume metadata.

Positional arguments:

<volume> ID of volume.

cinder metadata-update-all

```
usage: cinder metadata-update-all <volume> <key=value> [<key=value> ...]
```

Updates volume metadata.

Positional arguments:

<volume> ID of volume for which to update metadata.

<key=value> Metadata key and value pair or pairs to update.

cinder migrate

```
usage: cinder migrate [--force-host-copy [<True|False>]]
                      [--lock-volume [<True|False>]] [--cluster CLUSTER]
                      <volume> [<host>]
```

Migrates volume to a new host.

Positional arguments:

<volume> ID of volume to migrate.

<host> Destination host. Takes the form: host@backend-name#pool

Optional arguments:

--force-host-copy <True|False> Enables or disables generic host-based force-migration, which bypasses driver optimizations. Default=False.

--lock-volume <True|False> Enables or disables the termination of volume migration caused by other commands. This option applies to the available volume. True means it locks the volume state and does not allow the migration to be aborted. The volume status will be in maintenance during the migration. False means it allows the volume migration to be aborted. The volume status is still in the original status. Default=False.

--cluster None Destination cluster. Takes the form: cluster@backend-name#pool

cinder qos-associate

```
usage: cinder qos-associate <qos_specs> <volume_type_id>
```

Associates qos specs with specified volume type.

Positional arguments:

<qos_specs> ID of QoS specifications.

<volume_type_id> ID of volume type with which to associate QoS specifications.

cinder qos-create

```
usage: cinder qos-create <name> <key=value> [<key=value> ...]
```

Creates a qos specs.

Positional arguments:

<name> Name of new QoS specifications.

<key=value> QoS specifications.

cinder qos-delete

```
usage: cinder qos-delete [--force [<True|False>]] <qos_specs>
```

Deletes a specified qos specs.

Positional arguments:

<qos_specs> ID of QoS specifications to delete.

Optional arguments:

--force <True|False> Enables or disables deletion of in-use QoS specifications. Default=False.

cinder qos-disassociate

```
usage: cinder qos-disassociate <qos_specs> <volume_type_id>
```

Disassociates qos specs from specified volume type.

Positional arguments:

<qos_specs> ID of QoS specifications.

<volume_type_id> ID of volume type with which to associate QoS specifications.

cinder qos-disassociate-all

```
usage: cinder qos-disassociate-all <qos_specs>
```

Disassociates qos specs from all its associations.

Positional arguments:

<qos_specs> ID of QoS specifications on which to operate.

cinder qos-get-association

```
usage: cinder qos-get-association <qos_specs>
```

Lists all associations for specified qos specs.

Positional arguments:

<qos_specs> ID of QoS specifications.

cinder qos-key

```
usage: cinder qos-key <qos_specs> <action> key=value [key=value ...]
```

Sets or unsets specifications for a qos spec.

Positional arguments:

<qos_specs> ID of QoS specifications.

<action> The action. Valid values are set or unset.

key=value Metadata key and value pair to set or unset. For unset, specify only the key.

cinder qos-list

```
usage: cinder qos-list
```

Lists qos specs.

cinder qos-show

```
usage: cinder qos-show <qos_specs>
```

Shows qos specs details.

Positional arguments:

<qos_specs> ID of QoS specifications to show.

cinder quota-class-show

```
usage: cinder quota-class-show <class>
```

Lists quotas for a quota class.

Positional arguments:

<class> Name of quota class for which to list quotas.

cinder quota-class-update

```
usage: cinder quota-class-update [--volumes <volumes>]
                                 [--snapshots <snapshots>]
                                 [--gigabytes <gigabytes>]
                                 [--backups <backups>]
                                 [--backup-gigabytes <backup_gigabytes>]
                                 [--volume-type <volume_type_name>]
                                 [--per-volume-gigabytes <per_volume_
↪gigabytes>]
                                 <class_name>
```

Updates quotas for a quota class.

Positional arguments:

`<class_name>` Name of quota class for which to set quotas.

Optional arguments:

`--volumes <volumes>` The new volumes quota value. Default=None.

`--snapshots <snapshots>` The new snapshots quota value. Default=None.

`--gigabytes <gigabytes>` The new gigabytes quota value. Default=None.

`--backups <backups>` The new backups quota value. Default=None.

`--backup-gigabytes <backup_gigabytes>` The new backup_gigabytes quota value. Default=None.

`--volume-type <volume_type_name>` Volume type. Default=None.

`--per-volume-gigabytes <per_volume_gigabytes>` Set max volume size limit. Default=None.

cinder quota-defaults

```
usage: cinder quota-defaults <tenant_id>
```

Lists default quotas for a tenant.

Positional arguments:

`<tenant_id>` ID of tenant for which to list quota defaults.

cinder quota-delete

```
usage: cinder quota-delete <tenant_id>
```

Delete the quotas for a tenant.

Positional arguments:

`<tenant_id>` UUID of tenant to delete the quotas for.

cinder quota-show

```
usage: cinder quota-show <tenant_id>
```

Lists quotas for a tenant.

Positional arguments:

`<tenant_id>` ID of tenant for which to list quotas.

cinder quota-update

```
usage: cinder quota-update [--volumes <volumes>] [--snapshots <snapshots>]
                           [--gigabytes <gigabytes>] [--backups <backups>]
                           [--backup-gigabytes <backup_gigabytes>]
                           [--groups <groups>]
                           [--volume-type <volume_type_name>]
                           [--per-volume-gigabytes <per_volume_gigabytes>]
                           [--skip-validation <skip_validation>]
                           <tenant_id>
```

Updates quotas for a tenant.

Positional arguments:

<tenant_id> ID of tenant for which to set quotas.

Optional arguments:

--volumes <volumes> The new volumes quota value. Default=None.

--snapshots <snapshots> The new snapshots quota value. Default=None.

--gigabytes <gigabytes> The new gigabytes quota value. Default=None.

--backups <backups> The new backups quota value. Default=None.

--backup-gigabytes <backup_gigabytes> The new backup_gigabytes quota value. Default=None.

--groups <groups> The new groups quota value. Default=None.

--volume-type <volume_type_name> Volume type. Default=None.

--per-volume-gigabytes <per_volume_gigabytes> Set max volume size limit. Default=None.

--skip-validation <skip_validation> Skip validate the existing resource quota. Default=False.

cinder quota-usage

```
usage: cinder quota-usage <tenant_id>
```

Lists quota usage for a tenant.

Positional arguments:

<tenant_id> ID of tenant for which to list quota usage.

cinder rate-limits

```
usage: cinder rate-limits [<tenant_id>]
```

Lists rate limits for a user.

Positional arguments:

<tenant_id> Display information for a single tenant (Admin only).

cinder readonly-mode-update

```
usage: cinder readonly-mode-update <volume> <True|true|False|false>
```

Updates volume read-only access-mode flag.

Positional arguments:

<volume> ID of volume to update.

<True|true|False|false> Enables or disables update of volume to read-only access mode.

cinder reimage

```
usage: cinder reimage [--reimage-reserved <True|False>] <volume> <image-id>
```

Rebuilds a volume, overwriting all content with the specified image

Positional arguments:

<volume> Name or ID of volume to reimage

<image-id> The image id of the image that will be used to reimage the volume.

Optional arguments:

--reimage-reserved <True|False> Enables or disables reimage for a volume that is in reserved state otherwise only volumes in available or error status may be re-imaged. Default=False.

cinder rename

```
usage: cinder rename [--description <description>] <volume> [<name>]
```

Renames a volume.

Positional arguments:

<volume> Name or ID of volume to rename.

<name> New name for volume.

Optional arguments:

--description <description> Volume description. Default=None.

cinder reset-state

```
usage: cinder reset-state [--type <type>] [--state <state>]
                          [--attach-status <attach-status>]
                          [--reset-migration-status]
                          <entity> [<entity> ...]
```

Explicitly updates the entity state in the Cinder database.

Positional arguments:

<entity> Name or ID of entity to update.

Optional arguments:

--type <type> Type of entity to update. Available resources are: volume, snapshot, backup, group (since 3.20) and group-snapshot (since 3.19), Default=volume.

--state <state> The state to assign to the entity. NOTE: This command simply changes the state of the entity in the database with no regard to actual status, exercise caution when using. Default=None, that means the state is unchanged.

--attach-status <attach-status> This is only used for a volume entity. The attach status to assign to the volume in the database, with no regard to the actual status. Valid values are attached and detached. Default=None, that means the status is unchanged.

--reset-migration-status None This is only used for a volume entity. Clears the migration status of the volume in the DataBase that indicates the volume is source or destination of volume migration, with no regard to the actual status.

cinder retype

```
usage: cinder retype [--migration-policy <never|on-demand>]
                     <volume> <volume-type>
```

Changes the volume type for a volume.

Positional arguments:

<volume> Name or ID of volume for which to modify type.

<volume-type> New volume type.

Optional arguments:

--migration-policy <never|on-demand> Migration policy during retype of volume.

cinder revert-to-snapshot

```
usage: cinder revert-to-snapshot <snapshot>
```

Revert a volume to the specified snapshot.

Positional arguments:

<snapshot> Name or ID of the snapshot to restore. The snapshot must be the most recent one known to cinder.

cinder service-disable

```
usage: cinder service-disable [--reason <reason>] <hostname> <binary>
```

Disables the service.

Positional arguments:

<hostname> Host name.

<binary> Service binary.

Optional arguments:

--reason <reason> Reason for disabling service.

cinder service-enable

```
usage: cinder service-enable <hostname> <binary>
```

Enables the service.

Positional arguments:

<hostname> Host name.

<binary> Service binary.

cinder service-get-log

```
usage: cinder service-get-log
                              [--binary {,*,cinder-api,cinder-volume,cinder-
→scheduler,cinder-backup}]
                              [--server SERVER] [--prefix PREFIX]
```

Gets the service log level.

Optional arguments:

--binary None Binary to query.

--server None Host or cluster value for service.

--prefix None Prefix for the log. ie: sqlalchemy..

---

cinder service-list

```
usage: cinder service-list [--host <hostname>] [--binary <binary>]
                           [--withreplication [<True|False>]]
```

Lists all services. Filter by host and service binary.

Optional arguments:

--host <hostname> Host name. Default=None.

--binary <binary> Service binary. Default=None.

--withreplication <True|False> Enables or disables display of Replication info for c-vol services. Default=False.

cinder service-set-log

```
usage: cinder service-set-log
                             [--binary {,*,cinder-api,cinder-volume,cinder-
→scheduler,cinder-backup}]
                             [--server SERVER] [--prefix PREFIX]
                             <log-level>
```

Sets the service log level.

Positional arguments:

<log-level> Desired log level.

Optional arguments:

--binary None Binary to change.

--server None Host or cluster value for service.

--prefix None Prefix for the log. ie: cinder.volume.drivers..

cinder set-bootable

```
usage: cinder set-bootable <volume> <True|true|False|false>
```

Update bootable status of a volume.

Positional arguments:

<volume> ID of the volume to update.

<True|true|False|false> Flag to indicate whether volume is bootable.

cinder show

```
usage: cinder show <volume>
```

Shows volume details.

Positional arguments:

<volume> Name or ID of volume.

cinder snapshot-create

```
usage: cinder snapshot-create [--name <name>] [--description <description>]
                              [--metadata [<key=value> ...]]
                              <volume>
```

Creates a snapshot.

Positional arguments:

<volume> Name or ID of volume to snapshot.

Optional arguments:

--name <name> Snapshot name. Default=None.

--description <description> Snapshot description. Default=None.

--metadata <key=value> Snapshot metadata key and value pairs. Default=None.

cinder snapshot-delete

```
usage: cinder snapshot-delete [--force] <snapshot> [<snapshot> ...]
```

Removes one or more snapshots.

Positional arguments:

<snapshot> Name or ID of the snapshot(s) to delete.

Optional arguments:

--force None Allows deleting snapshot of a volume when its status is other than available or error.
Default=False.

cinder snapshot-list

```
usage: cinder snapshot-list [--all-tenants [<0|1>]] [--name <name>]
                            [--status <status>] [--volume-id <volume-id>]
                            [--marker <marker>] [--limit <limit>]
                            [--sort <key>[:<direction>]] [--tenant [<tenant>]]
                            [--metadata [<key=value> ...]]
                            [--filters [<key=value> ...]]
                            [--with-count [<True|False>]]
```

Lists all snapshots.

Optional arguments:

`--all-tenants <0|1>` Shows details for all tenants. Admin only.

`--name <name>` Filters results by a name. Default=None. This option is deprecated and will be removed in newer release. Please use filters option which is introduced since 3.33 instead.

`--status <status>` Filters results by a status. Default=None. This option is deprecated and will be removed in newer release. Please use filters option which is introduced since 3.33 instead.

`--volume-id <volume-id>` Filters results by a volume ID. Default=None. This option is deprecated and will be removed in newer release. Please use filters option which is introduced since 3.33 instead.

`--marker <marker>` Begin returning snapshots that appear later in the snapshot list than that represented by this id. Default=None.

`--limit <limit>` Maximum number of snapshots to return. Default=None.

`--sort <key>[:<direction>]` Comma-separated list of sort keys and directions in the form of <key>[:<asc|desc>]. Valid keys: id, status, size, availability_zone, name, bootable, created_at, reference. Default=None.

`--tenant <tenant>` Display information from single tenant (Admin only).

`--metadata <key=value>` Filters results by a metadata key and value pair. Require volume api version >=3.22. Default=None. This option is deprecated and will be removed in newer release. Please use filters option which is introduced since 3.33 instead.

`--filters <key=value>` Filter key and value pairs. Please use cinder list-filters to check enabled filters from server. Use key~=value for inexact filtering if the key supports. Default=None.

`--with-count <True|False>` Show total number of snapshot entities. This is useful when pagination is applied in the request.

cinder snapshot-manage

```
usage: cinder snapshot-manage [--id-type <id-type>] [--name <name>]
                              [--description <description>]
                              [--metadata [<key=value> ...]]
                              <volume> <identifier>
```

Manage an existing snapshot.

Positional arguments:

`<volume>` Cinder volume that already exists in the volume backend.

`<identifier>` Name or other identifier for existing snapshot. This is backend specific.

Optional arguments:

`--id-type <id-type>` Type of backend device identifier provided, typically source-name or source-id (Default=source-name).

`--name <name>` Snapshot name (Default=None).

`--description <description>` Snapshot description (Default=None).

`--metadata <key=value>` Metadata key=value pairs (Default=None).

cinder snapshot-manageable-list

```
usage: cinder snapshot-manageable-list [--detailed <detailed>]
                                       [--marker <marker>] [--limit <limit>]
                                       [--offset <offset>]
                                       [--sort <key>[:<direction>]]
                                       [--cluster CLUSTER]
                                       [<host>]
```

Lists all manageable snapshots.

Positional arguments:

`<host>` Cinder host on which to list manageable snapshots; takes the form: host@backend-name#pool

Optional arguments:

`--detailed <detailed>` Returned detailed information (default true).

`--marker <marker>` Begin returning volumes that appear later in the volume list than that represented by this reference. This reference should be json like. Default=None.

`--limit <limit>` Maximum number of volumes to return. Default=None.

`--offset <offset>` Number of volumes to skip after marker. Default=None.

`--sort <key>[:<direction>]` Comma-separated list of sort keys and directions in the form of <key>[:<asc|desc>]. Valid keys: size, reference. Default=None.

`--cluster None` Cinder cluster on which to list manageable snapshots; takes the form: cluster@backend-name#pool

cinder snapshot-metadata

```
usage: cinder snapshot-metadata <snapshot> <action> <key=value>
                                [<key=value> ...]
```

Sets or deletes snapshot metadata.

Positional arguments:

`<snapshot>` ID of snapshot for which to update metadata.

`<action>` The action. Valid values are set or unset.

`<key=value>` Metadata key and value pair to set or unset. For unset, specify only the key.

cinder snapshot-metadata-show

```
usage: cinder snapshot-metadata-show <snapshot>
```

Shows snapshot metadata.

Positional arguments:

<snapshot> ID of snapshot.

cinder snapshot-metadata-update-all

```
usage: cinder snapshot-metadata-update-all <snapshot> <key=value>
                                            [<key=value> ...]
```

Updates snapshot metadata.

Positional arguments:

<snapshot> ID of snapshot for which to update metadata.

<key=value> Metadata key and value pair to update.

cinder snapshot-rename

```
usage: cinder snapshot-rename [--description <description>]
                              <snapshot> [<name>]
```

Renames a snapshot.

Positional arguments:

<snapshot> Name or ID of snapshot.

<name> New name for snapshot.

Optional arguments:

--description <description> Snapshot description. Default=None.

cinder snapshot-reset-state

```
usage: cinder snapshot-reset-state [--state <state>]
                                   <snapshot> [<snapshot> ...]
```

Explicitly updates the snapshot state.

Positional arguments:

<snapshot> Name or ID of snapshot to modify.

Optional arguments:

--state <state> The state to assign to the snapshot. Valid values are available, error, creating, deleting, and error_deleting. NOTE: This command simply changes the state of the Snapshot in the DataBase with no regard to actual status, exercise caution when using. Default=available.

cinder snapshot-show

```
usage: cinder snapshot-show <snapshot>
```

Shows snapshot details.

Positional arguments:

<snapshot> Name or ID of snapshot.

cinder snapshot-unmanage

```
usage: cinder snapshot-unmanage <snapshot>
```

Stop managing a snapshot.

Positional arguments:

<snapshot> Name or ID of the snapshot to unmanage.

cinder summary

```
usage: cinder summary [--all-tenants [<0|1>]]
```

Get volumes summary.

Optional arguments:

--all-tenants <0|1> Shows details for all tenants. Admin only.

cinder thaw-host

```
usage: cinder thaw-host <hostname>
```

Thaw and enable the specified cinder-volume host.

Positional arguments:

<hostname> Host name.

cinder transfer-accept

```
usage: cinder transfer-accept <transfer> <auth_key>
```

Accepts a volume transfer.

Positional arguments:

<transfer> ID of transfer to accept.

<auth_key> Authentication key of transfer to accept.

cinder transfer-create

```
usage: cinder transfer-create [--name <name>] [--no-snapshots] <volume>
```

Creates a volume transfer.

Positional arguments:

<volume> Name or ID of volume to transfer.

Optional arguments:

--name <name> Transfer name. Default=None.

--no-snapshots None Allows or disallows transfer volumes without snapshots. Default=False.

cinder transfer-delete

```
usage: cinder transfer-delete <transfer>
```

Undoes a transfer.

Positional arguments:

<transfer> Name or ID of transfer to delete.

cinder transfer-list

```
usage: cinder transfer-list [--all-tenants [<0|1>]]
                            [--sort <key>[:<direction>]]
                            [--filters [<key=value> ...]]
```

Lists all transfers.

Optional arguments:

--all-tenants <0|1> Shows details for all tenants. Admin only.

--sort <key>[:<direction>] Sort keys and directions in the form of <key>[:<asc|desc>].

--filters <key=value> Filter key and value pairs.

cinder transfer-show

```
usage: cinder transfer-show <transfer>
```

Shows transfer details.

Positional arguments:

<transfer> Name or ID of transfer to accept.

cinder type-access-add

```
usage: cinder type-access-add --volume-type <volume_type> --project-id
                              <project_id>
```

Adds volume type access for the given project.

Optional arguments:

--volume-type <volume_type> Volume type name or ID to add access for the given project.

--project-id <project_id> Project ID to add volume type access for.

cinder type-access-list

```
usage: cinder type-access-list --volume-type <volume_type>
```

Print access information about the given volume type.

Optional arguments:

--volume-type <volume_type> Filter results by volume type name or ID.

cinder type-access-remove

```
usage: cinder type-access-remove --volume-type <volume_type> --project-id
                                 <project_id>
```

Removes volume type access for the given project.

Optional arguments:

--volume-type <volume_type> Volume type name or ID to remove access for the given project.

--project-id <project_id> Project ID to remove volume type access for.

cinder type-create

```
usage: cinder type-create [--description <description>]
                          [--is-public <is-public>]
                          <name>
```

Creates a volume type.

Positional arguments:

<name> Name of new volume type.

Optional arguments:

--description <description> Description of new volume type.

--is-public <is-public> Make type accessible to the public (default true).

cinder type-default

```
usage: cinder type-default
```

List the default volume type.

cinder type-delete

```
usage: cinder type-delete <vol_type> [<vol_type> ...]
```

Deletes volume type or types.

Positional arguments:

<vol_type> Name or ID of volume type or types to delete.

cinder type-key

```
usage: cinder type-key <vtype> <action> <key=value> [<key=value> ...]
```

Sets or unsets extra_spec for a volume type.

Positional arguments:

<vtype> Name or ID of volume type.

<action> The action. Valid values are set or unset.

<key=value> The extra specs key and value pair to set or unset. For unset, specify only the key.

cinder type-list

```
usage: cinder type-list [--filters [<key=value> ...]]
```

Lists available volume types.

Optional arguments:

--filters <key=value> Filter key and value pairs. Admin only.

cinder type-show

```
usage: cinder type-show <volume_type>
```

Show volume type details.

Positional arguments:

<volume_type> Name or ID of the volume type.

cinder type-update

```
usage: cinder type-update [--name <name>] [--description <description>]
                          [--is-public <is-public>]
                          <id>
```

Updates volume type name, description, and/or is_public.

Positional arguments:

<id> ID of the volume type.

Optional arguments:

--name <name> Name of the volume type.

--description <description> Description of the volume type.

--is-public <is-public> Make type accessible to the public or not.

cinder unmanage

```
usage: cinder unmanage <volume>
```

Stop managing a volume.

Positional arguments:

<volume> Name or ID of the volume to unmanage.

cinder upload-to-image

```
usage: cinder upload-to-image [--force [<True|False>]]
                              [--container-format <container-format>]
                              [--disk-format <disk-format>]
                              [--visibility <public|private|community|shared>]
                              [--protected <True|False>]
                              <volume> <image-name>
```

Uploads volume to Image Service as an image.

Positional arguments:

`<volume>` Name or ID of volume to snapshot.

`<image-name>` The new image name.

Optional arguments:

`--force <True|False>` Enables or disables upload of a volume that is attached to an instance. Default=False. This option may not be supported by your cloud.

`--container-format <container-format>` Container format type. Default is bare.

`--disk-format <disk-format>` Disk format type. Default is raw.

`--visibility <public|private|community|shared>` Set image visibility to public, private, community or shared. Default=private.

`--protected <True|False>` Prevents image from being deleted. Default=False.

cinder version-list

```
usage: cinder version-list
```

List all API versions.

cinder work-cleanup

```
usage: cinder work-cleanup [--cluster <cluster-name>] [--host <hostname>]
                           [--binary <binary>]
                           [--is-up <True|true|False|false>]
                           [--disabled <True|true|False|false>]
                           [--resource-id <resource-id>]
                           [--resource-type <Volume|Snapshot>]
                           [--service-id <service-id>]
```

Request cleanup of services with optional filtering.

Optional arguments:

`--cluster <cluster-name>` Cluster name. Default=None.

`--host <hostname>` Service host name. Default=None.

`--binary <binary>` Service binary. Default=None.

--is-up <True|true|False|false> Filter by up/down status, if set to true services need to be up, if set to false services need to be down. Default is None, which means up/down status is ignored.

--disabled <True|true|False|false> Filter by disabled status. Default=None.

--resource-id <resource-id> UUID of a resource to cleanup. Default=None.

--resource-type <Volume|Snapshot> Type of resource to cleanup.

--service-id <service-id> The service id field from the DB, not the uuid of the service. Default=None.

cinder help

```
usage: cinder help [<subcommand>]
```

Shows help about this program or one of its subcommands.

Positional arguments:

<subcommand> Shows help for <subcommand>.

DEVELOPER GUIDES

3.1 So You Want to Contribute

For general information on contributing to OpenStack, please check out the contributor guide to get started. It covers all the basics that are common to all OpenStack projects: the accounts you need, the basics of interacting with our Gerrit review system, how we communicate as a community, etc.

The python-cinderclient is maintained by the OpenStack Cinder project. To understand our development process and how you can contribute to it, please look at the Cinder projects general contributors page: http://docs.openstack.org/cinder/latest/contributor/contributing.html

3.2 Functional Tests

Cinderclient contains a suite of functional tests, in the cinderclient/ tests/functional directory.

These are currently non-voting, meaning that zuul will not reject a patched based on failure of the functional tests. It is highly recommended, however, that these tests are investigated in the case of a failure.

3.2.1 Running the tests

Run the tests using tox, via the tox.ini file. To run all tests simply run:

```
tox -e functional
```

This will create a virtual environment, load all the packages from test-requirements.txt and run all unit tests as well as run flake8 and hacking checks against the code.

Note that you can inspect the tox.ini file to get more details on the available options and what the test run does by default.

3.2.2 Running a subset of tests using tox

One common activity is to just run a single test, you can do this with tox simply by specifying to just run py27 or py34 tests against a single test:

```
tox -e functional -- -n cinderclient.tests.functional.test_readonly_cli.
↪CinderClientReadOnlyTests.test_list
```

Or all tests in the test_readonly_clitest_readonly_cli.py file:

```
tox -e functional -- -n cinderclient.tests.functional.test_readonly_cli
```

For more information on these options and how to run tests, please see the stestr documentation.

3.2.3 Gotchas

The cinderclient.tests.functional.test_cli.CinderBackupTests.test_backup_create and_delete test will fail in Devstack without c-bak service running, which requires Swift. Make sure Swift is enabled when you stack.sh by putting this in local.conf:

```
enable_service s-proxy s-object s-container s-account
```

3.3 Unit Tests

Cinderclient contains a suite of unit tests, in the cinderclient/tests/unit directory.

Any proposed code change will be automatically rejected by the OpenStack Jenkins server if the change causes unit test failures.

3.3.1 Running the tests

There are a number of ways to run unit tests currently, and theres a combination of frameworks used depending on what commands you use. The preferred method is to use tox, which calls ostestr via the tox.ini file. To run all tests simply run:

```
tox
```

This will create a virtual environment, load all the packages from test-requirements.txt and run all unit tests as well as run flake8 and hacking checks against the code.

Note that you can inspect the tox.ini file to get more details on the available options and what the test run does by default.

3.3.2 Running a subset of tests using tox

One common activity is to just run a single test, you can do this with tox simply by specifying to just run py3 tests against a single test:

```
tox -e py3 -- -n cinderclient.tests.unit.v3.test_volumes.VolumesTest.test_
↪create_volume
```

Or all tests in the test_volumes.py file:

```
tox -e py3 -- -n cinderclient.tests.unit.v3.test_volumes
```

For more information on these options and how to run tests, please see the stestr documentation.

3.3.3 Gotchas

Running Tests from Shared Folders

If you are running the unit tests from a shared folder, you may see tests start to fail or stop completely as a result of Python lockfile issues. You can get around this by manually setting or updating the following line in `cinder/tests/conf_fixture.py`:

```
CONF['lock_path'].SetDefault('/tmp')
```

Note that you may use any location (not just `/tmp`!) as long as it is not a shared folder.

RELEASE NOTES

All python-cinderclient release notes can now be found on the release notes page.

The following are kept for historical purposes.

4.1 1.4.0

- Improved error reporting on reaching quota.

- Volume status management for volume migration.

- Added command to fetch specified backend capabilities.

- Added commands for modifying image metadata.

- Support for non-disruptive backup.

- Support for cloning consistency groups.

4.2 1.3.1

- Fixed usage of the debug option.

- Documentation and API example improvements.

- Set max volume size limit for the project.

- Added encryption-type-update to cinderclient.

- Added volume multi attach support.

- Support host-attach of volumes.

4.3 1.3.0

- Revert version discovery support due to this breaking deployments using proxies. We will revisit this once the Kilo config option public_endpoint has been available longer to allow these deployments to work again with version discovery available from the Cinder client.

- Add volume multi-attach support.

- Add encryption-type-update to update volume encryption types.

4.4 1.2.2

- IMPORTANT: version discovery breaks deployments using proxies and has been reverted in v1.3.0
  . Do not use this version.

- Update requirements to resolve conflicts with other OpenStack projects

4.5 1.2.1

- IMPORTANT: version discovery breaks deployments using proxies and has been reverted in v1.3.0
  . Do not use this version.

- Remove warnings about Keystone unable to contact endpoint for discovery.

- backup-create subcommand allows specifying incremental to do an incremental backup.

- Modify consistency groups using the consisgroup-update subcommand. Change the name, description, add volumes, or remove volumes.

- Create consistency group from consistency group snapshot using the consisgroup-create-from-src subcommand.

- force no longer needs a boolean to be specified.

4.6 1.2.0

- IMPORTANT: version discovery breaks deployments using proxies and has been reverted in v1.3.0
  . Do not use this version.

- Add metadata during snapshot create.

- Add TTY password entry when no password is environment vars or os-password.

- Ability to set backup quota in quota-update subcommand.

- Force the client to use a particular Cinder API endpoint with bypass-url.

- Create a volume from an image by image name.

- New type-default subcommand will display the default volume type.

- New type-update subcommand allows updating a volume types description.

- type-list subcommand displays volume type description.

- type-create subcommand allows setting the description.

- Show pools to a backend when doing a service-list subcommand.

- List and update consistency group quotas.

- Create volume types that are non-public and have particular project access.

- -d is available as a shorter option to debug.

- transfer-list subcommand has an option for all-tenants.

- sort option available instead of sort-key and sort-dir. E.q. sort <key>[:<direction>].

- Volume type name can now be updated via subcommand type-update.

- bash completion gives subcommands when using cinder help.

- Version discovery is now available. You no longer need a volumev2 service type in your keystone catalog.

- Filter by tenant in list subcommand.

** Python 2.4 support removed.

** sort-key and sort-dir are deprecated. Use sort instead.

** **A dash will be displayed of None when there is no data to display under** a column.

4.7 1.1.1

4.8 1.1.0

- Add support for ConsistencyGroups

- Use Adapter from keystoneclient

- Add support for Replication feature

- Add pagination for Volume List

- Note Connection refused > Connection error commit: c9e7818f3f90ce761ad8ccd09181c705880a4266

- Note Mask Passwords in log output commit: 80582f2b860b2dadef7ae07bdbd8395bf03848b1

4.9 1.0.9

4.10 1.0.8

- Add support for reset-state on multiple volumes or snapshots at once

- Add volume retype command

4.11 1.0.7

- Add support for read-only volumes

- Add support for setting snapshot metadata

- Deprecate volume-id arg to backup restore in favor of volume

- Add quota-usage command

- Fix exception deprecation warning message

- Report error when no args supplied to rename cmd

4.12 1.0.6

- Add support for multiple endpoints

- Add response info for backup command

- Add metadata option to cinder list command

- Add timeout parameter for requests

- Add update action for snapshot metadata

- Add encryption metadata support

- Add volume migrate support

- Add support for QoS specs

4.13 1.0.5

- Add CLI man page

- Add Availability Zone list command

- Add support for scheduler-hints

- Add support to extend volumes

- Add support to reset state on volumes and snapshots

- Add snapshot support for quota class

4.14 1.0.4

- Added support for backup-service commands

4.15 1.0.3

- Added support for V2 Cinder API

- Corrected upload-volume-to-image help messaging

- Align handling of metadata args for all methods

- Update OSLO version

- Correct parsing of volume metadata

- Enable force delete of volumes and snapshots in error state

- Implement clone volume API call

- Add list-extensions call to cinderclient

- Add bootable column to list output

- Add retries to cinderclient operations

- Add Type/Extra-Specs support

- Add volume and snapshot rename commands