

ADMIN GUIDE

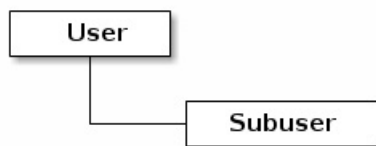
Once you have your Ceph Object Storage service up and running, you may administer the service with user management, access controls, quotas and usage tracking among other features.

USER MANAGEMENT

Ceph Object Storage user management refers to users of the Ceph Object Storage service (i.e., not the Ceph Object Gateway as a user of the Ceph Storage Cluster). You must create a user, access key and secret to enable end users to interact with Ceph Object Gateway services.

There are two user types:

- **User:** The term 'user' reflects a user of the S3 interface.
- **Subuser:** The term 'subuser' reflects a user of the Swift interface. A subuser is associated to a user .



You can create, modify, view, suspend and remove users and subusers. In addition to user and subuser IDs, you may add a display name and an email address for a user. You can specify a key and secret, or generate a key and secret automatically. When generating or specifying keys, note that user IDs correspond to an S3 key type and subuser IDs correspond to a swift key type. Swift keys also have access levels of read, write, readwrite and full.

CREATE A USER

To create a user (S3 interface), execute the following:

```
radosgw-admin user create --uid={username} --display-name="{display-name}" [--email={email}]
```

For example:

```
radosgw-admin user create --uid=johndoe --display-name="John Doe" --email=john@example.com
```

```
{ "user_id": "johndoe",
  "display_name": "John Doe",
  "email": "john@example.com",
  "suspended": 0,
  "max_buckets": 1000,
  "auid": 0,
  "subusers": [],
  "keys": [
    { "user": "johndoe",
      "access_key": "11BS02LGFB6AL6H1ADMW",
      "secret_key": "vzCEkuryfn060dfee4fgQPqFrncKEIkh3Zcd0ANY"}],
  "swift_keys": [],
  "caps": [],
  "op_mask": "read, write, delete",
  "default_placement": "",
  "placement_tags": [],
  "bucket_quota": { "enabled": false,
    "max_size_kb": -1,
    "max_objects": -1},
  "user_quota": { "enabled": false,
    "max_size_kb": -1,
    "max_objects": -1},
  "temp_url_keys": []}
```

Creating a user also creates an `access_key` and `secret_key` entry for use with any S3 API-compatible client.

Important: Check the key output. Sometimes `radosgw-admin` generates a JSON escape (`\`) character, and some clients do not know how to handle JSON escape characters. Remedies include removing the JSON escape character (`\`), encapsulating the string in quotes, regenerating the key and ensuring that it does not have a JSON escape character or specify the key and secret manually.

CREATE A SUBUSER

To create a subuser (Swift interface) for the user, you must specify the user ID (`--uid={username}`), a subuser ID and the access level for the subuser.

```
radosgw-admin subuser create --uid={uid} --subuser={uid} --access=[ read | write | readwrite
```

For example:

```
radosgw-admin subuser create --uid=johndoe --subuser=johndoe:swift --access=full
```

Note: `full` is not `readwrite`, as it also includes the access control policy.

```
{ "user_id": "johndoe",
  "display_name": "John Doe",
  "email": "john@example.com",
  "suspended": 0,
  "max_buckets": 1000,
  "auid": 0,
  "subusers": [
    { "id": "johndoe:swift",
      "permissions": "full-control" }],
  "keys": [
    { "user": "johndoe",
      "access_key": "11BS02LGFB6AL6H1ADMW",
      "secret_key": "vzCEkuryfn060dfee4fgQPqFrncKEIkh3Zcd0ANY" }],
  "swift_keys": [],
  "caps": [],
  "op_mask": "read, write, delete",
  "default_placement": "",
  "placement_tags": [],
  "bucket_quota": { "enabled": false,
    "max_size_kb": -1,
    "max_objects": -1 },
  "user_quota": { "enabled": false,
    "max_size_kb": -1,
    "max_objects": -1 },
  "temp_url_keys": [] }
```

GET USER INFO

To get information about a user, you must specify `user info` and the user ID (`--uid={username}`).

```
radosgw-admin user info --uid=johndoe
```

MODIFY USER INFO

To modify information about a user, you must specify the user ID (`--uid={username}`) and the attributes you want to modify. Typical modifications are to keys and secrets, email addresses, display names and access levels. For example:

```
radosgw-admin user modify --uid=johndoe --display-name="John E. Doe"
```

To modify subuser values, specify `subuser modify` and the subuser ID. For example:

```
radosgw-admin subuser modify --subuser=johndoe:swift --access=full
```

USER ENABLE/SUSPEND

When you create a user, the user is enabled by default. However, you may suspend user privileges and re-enable them at a later time. To suspend a user, specify `user suspend` and the user ID.

```
radosgw-admin user suspend --uid=johndoe
```

To re-enable a suspended user, specify `user enable` and the user ID.

```
radosgw-admin user enable --uid=johndoe
```

Note: Disabling the user disables the subuser.

REMOVE A USER

When you remove a user, the user and subuser are removed from the system. However, you may remove just the subuser if you wish. To remove a user (and subuser), specify `user rm` and the user ID.

```
radosgw-admin user rm --uid=johndoe
```

To remove the subuser only, specify `subuser rm` and the subuser ID.

```
radosgw-admin subuser rm --subuser=johndoe:swift
```

Options include:

- **Purge Data:** The `--purge-data` option purges all data associated to the UID.
- **Purge Keys:** The `--purge-keys` option purges all keys associated to the UID.

REMOVE A SUBUSER

When you remove a sub user, you are removing access to the Swift interface. The user will remain in the system. To remove the subuser, specify `subuser rm` and the subuser ID.

```
radosgw-admin subuser rm --subuser=johndoe:swift
```

Options include:

- **Purge Keys:** The `--purge-keys` option purges all keys associated to the UID.

ADD / REMOVE A KEY

Both users and subusers require the key to access the S3 or Swift interface. To use S3, the user needs a key pair which is composed of an access key and a secret key. On the other hand, to use Swift, the user typically needs a secret key (password), and use it together with the associated user ID. You may create a key and either specify or generate the access key and/or secret key. You may also remove a key. Options include:

- `--key-type=<type>` specifies the key type. The options are: `s3`, `swift`
- `--access-key=<key>` manually specifies an S3 access key.
- `--secret-key=<key>` manually specifies a S3 secret key or a Swift secret key.
- `--gen-access-key` automatically generates a random S3 access key.

- `--gen-secret` automatically generates a random S3 secret key or a random Swift secret key.

An example how to add a specified S3 key pair for a user.

```
radosgw-admin key create --uid=foo --key-type=s3 --access-key fooAccessKey --secret-key fooSe
```

```
{ "user_id": "foo",
  "rados_uid": 0,
  "display_name": "foo",
  "email": "foo@example.com",
  "suspended": 0,
  "keys": [
    { "user": "foo",
      "access_key": "fooAccessKey",
      "secret_key": "fooSecretKey"}],
}
```

Note that you may create multiple S3 key pairs for a user.

To attach a specified swift secret key for a subuser.

```
radosgw-admin key create --subuser=foo:bar --key-type=swift --secret-key barSecret
```

```
{ "user_id": "foo",
  "rados_uid": 0,
  "display_name": "foo",
  "email": "foo@example.com",
  "suspended": 0,
  "subusers": [
    { "id": "foo:bar",
      "permissions": "full-control"}],
  "swift_keys": [
    { "user": "foo:bar",
      "secret_key": "asfghjghghmgm"}]}
```

Note that a subuser can have only one swift secret key.

Subusers can also be used with S3 APIs if the subuser is associated with a S3 key pair.

```
radosgw-admin key create --subuser=foo:bar --key-type=s3 --access-key barAccessKey --secret-k
```

```
{ "user_id": "foo",
  "rados_uid": 0,
  "display_name": "foo",
  "email": "foo@example.com",
  "suspended": 0,
  "subusers": [
    { "id": "foo:bar",
      "permissions": "full-control"}],
  "keys": [
    { "user": "foo:bar",
      "access_key": "barAccessKey",
      "secret_key": "barSecretKey"}],
}
```

To remove a S3 key pair, specify the access key.

```
radosgw-admin key rm --uid=foo --key-type=s3 --access-key=fooAccessKey
```

To remove the swift secret key.

```
radosgw-admin key rm --subuser=foo:bar --key-type=swift
```

ADD / REMOVE ADMIN CAPABILITIES

The Ceph Storage Cluster provides an administrative API that enables users to execute administrative functions via the REST API. By default, users do NOT have access to this API. To enable a user to exercise administrative functionality, provide the user with administrative capabilities.

To add administrative capabilities to a user, execute the following:

```
radosgw-admin caps add --uid={uid} --caps={caps}
```

You can add read, write or all capabilities to users, buckets, metadata and usage (utilization). For example:

```
--caps="[users|buckets|metadata|usage|zone]=[*|read|write|read, write]"
```

For example:

```
radosgw-admin caps add --uid=johndoe --caps="users=*;buckets=*"
```

To remove administrative capabilities from a user, execute the following:

```
radosgw-admin caps rm --uid=johndoe --caps={caps}
```

QUOTA MANAGEMENT

The Ceph Object Gateway enables you to set quotas on users and buckets owned by users. Quotas include the maximum number of objects in a bucket and the maximum storage size a bucket can hold.

- **Bucket:** The `--bucket` option allows you to specify a quota for buckets the user owns.
- **Maximum Objects:** The `--max-objects` setting allows you to specify the maximum number of objects. A negative value disables this setting.
- **Maximum Size:** The `--max-size` option allows you to specify a quota size in B/K/M/G/T, where B is the default. A negative value disables this setting.
- **Quota Scope:** The `--quota-scope` option sets the scope for the quota. The options are bucket and user. Bucket quotas apply to buckets a user owns. User quotas apply to a user.

SET USER QUOTA

Before you enable a quota, you must first set the quota parameters. For example:

```
radosgw-admin quota set --quota-scope=user --uid=<uid> [--max-objects=<num objects>] [--max-s
```

For example:

```
radosgw-admin quota set --quota-scope=user --uid=johndoe --max-objects=1024 --max-size=1024B
```

A negative value for num objects and / or max size means that the specific quota attribute check is disabled.

ENABLE/DISABLE USER QUOTA

Once you set a user quota, you may enable it. For example:

```
radosgw-admin quota enable --quota-scope=user --uid=<uid>
```

You may disable an enabled user quota. For example:

```
radosgw-admin quota disable --quota-scope=user --uid=<uid>
```

SET BUCKET QUOTA

Bucket quotas apply to the buckets owned by the specified uid. They are independent of the user.

```
radosgw-admin quota set --uid=<uid> --quota-scope=bucket [--max-objects=<num objects>] [--max
```

A negative value for num objects and / or max size means that the specific quota attribute check is disabled.

ENABLE/DISABLE BUCKET QUOTA

Once you set a bucket quota, you may enable it. For example:

```
radosgw-admin quota enable --quota-scope=bucket --uid=<uid>
```

You may disable an enabled bucket quota. For example:

```
radosgw-admin quota disable --quota-scope=bucket --uid=<uid>
```

GET QUOTA SETTINGS

You may access each user's quota settings via the user information API. To read user quota setting information with the CLI interface, execute the following:

```
radosgw-admin user info --uid=<uid>
```

UPDATE QUOTA STATS

Quota stats get updated asynchronously. You can update quota statistics for all users and all buckets manually to retrieve the latest quota stats.

```
radosgw-admin user stats --uid=<uid> --sync-stats
```

GET USER USAGE STATS

To see how much of the quota a user has consumed, execute the following:

```
radosgw-admin user stats --uid=<uid>
```

Note: You should execute `radosgw-admin user stats` with the `--sync-stats` option to receive the latest data.

DEFAULT QUOTAS

You can set default quotas in the config. These defaults are used when creating a new user and have no effect on existing users. If the relevant default quota is set in config, then that quota is set on the new user, and that quota is enabled. See `rgw`

bucket default quota max objects, rgw bucket default quota max size, rgw user default quota max objects, and rgw user default quota max size in [Ceph Object Gateway Config Reference](#)

QUOTA CACHE

Quota statistics are cached on each RGW instance. If there are multiple instances, then the cache can keep quotas from being perfectly enforced, as each instance will have a different view of quotas. The options that control this are `rgw bucket quota ttl`, `rgw user quota bucket sync interval` and `rgw user quota sync interval`. The higher these values are, the more efficient quota operations are, but the more out-of-sync multiple instances will be. The lower these values are, the closer to perfect enforcement multiple instances will achieve. If all three are 0, then quota caching is effectively disabled, and multiple instances will have perfect quota enforcement. See [Ceph Object Gateway Config Reference](#)

READING / WRITING GLOBAL QUOTAS

You can read and write global quota settings in the period configuration. To view the global quota settings:

```
radosgw-admin global quota get
```

The global quota settings can be manipulated with the `global quota` counterparts of the `quota set`, `quota enable`, and `quota disable` commands.

```
radosgw-admin global quota set --quota-scope bucket --max-objects 1024
radosgw-admin global quota enable --quota-scope bucket
```

Note: In a multisite configuration, where there is a realm and period present, changes to the global quotas must be committed using `period update --commit`. If there is no period present, the rados gateway(s) must be restarted for the changes to take effect.

USAGE

The Ceph Object Gateway logs usage for each user. You can track user usage within date ranges too.

- Add `rgw enable usage log = true` in `[client.rgw]` section of `ceph.conf` and restart the `radosgw` service.

Options include:

- **Start Date:** The `--start-date` option allows you to filter usage stats from a particular start date (**format:** `yyyy-mm-dd[HH:MM:SS]`).
- **End Date:** The `--end-date` option allows you to filter usage up to a particular date (**format:** `yyyy-mm-dd[HH:MM:SS]`).
- **Log Entries:** The `--show-log-entries` option allows you to specify whether or not to include log entries with the usage stats (options: `true` | `false`).

Note: You may specify time with minutes and seconds, but it is stored with 1 hour resolution.

SHOW USAGE

To show usage statistics, specify the `usage show`. To show usage for a particular user, you must specify a user ID. You may also specify a start date, end date, and whether or not to show log entries.:

```
radosgw-admin usage show --uid=johndoe --start-date=2012-03-01 --end-date=2012-04-01
```

You may also show a summary of usage information for all users by omitting a user ID.

```
radosgw-admin usage show --show-log-entries=false
```

TRIM USAGE

With heavy use, usage logs can begin to take up storage space. You can trim usage logs for all users and for specific users. You may also specify date ranges for trim operations.

```
radosgw-admin usage trim --start-date=2010-01-01 --end-date=2010-12-31
radosgw-admin usage trim --uid=johndoe
radosgw-admin usage trim --uid=johndoe --end-date=2013-12-31
```