Documentation

MinIO Client

The MinIO Client `mc` command line tool provides a modern alternative to UNIX commands like `ls`, `cat`, `cp`, `mirror`, and `diff` with support for both filesystems and Amazon S3-compatible cloud storage services.

The **mc** commandline tool is built for compatibility with the AWS S3 API and is tested with MinIO and AWS S3 for expected functionality and behavior.

MinIO provides no guarantees for other S3-compatible services, as their S3 API implementation is unknown and therefore unsupported. While **mc** commands *may* work as documented, any such usage is at your own risk.

`mc` has the following syntax:

```
mc [GLOBALFLAGS] COMMAND --help
```

See Command Quick Reference for a list of supported commands.

> **AGPLv3:**
>
> **mc** is AGPLv3 licensed Free and Open Source (FOSS) software.
>
> Applications integrating **mc** may trigger AGPLv3 compliance requirements. MinIO Commercial Licensing is the best option for applications which trigger AGPLv3 obligations where open-sourcing the application is not an option.

Quickstart

1) Install `mc`

Install the **mc** command line tool onto the host machine. Click the tab that corresponds to the host machine operating system or environment:

Linux macOS Windows Source

```
brew install minio/stable/mc
mc --help
```

Important:

The following example temporarily disables the bash history to mitigate the risk of authentication credentials leaking in plain text. This is a basic security measure and does not mitigate all possible attack vectors. Defer to security best practices for your operating system for inputting sensitive information on the command line.

Use the `mc alias set` command to add an Amazon S3-compatible service to the `mc` configuration.

```
bash +o history
mc alias set ALIAS HOSTNAME ACCESS_KEY SECRET_KEY
bash -o history
```

- Replace `ALIAS` with a name to associate to the S3 service. `mc` commands typically require `ALIAS` as an argument for identifying which S3 service to execute against.
- Replace `HOSTNAME` with the URL endpoint or IP address of the S3 service.
- Replace `ACCESS_KEY` and `SECRET_KEY` with the access and secret keys for a user on the S3 service.

Replace each argument with the required values. If you omit the `ACCESS_KEY` and `SECRET_KEY`, the command prompts you to enter those values in the CLI.

Each of the following tabs contains a provider-specific example:

MinIO Server   AWS S3 Storage   Google Cloud Storage

```
mc alias set myminio https://minioserver.example.net ACCESS_KEY SECRET KEY
```

3) Test the Connection

Use the `mc admin info` command to test the connection to the newly added MinIO deployment:

```
mc admin info myminio
```

The command returns information on the S3 service if successful. If unsuccessful, check each of the following:

- The host machine has connectivity to the S3 service URL (i.e. using `ping` or

- The specified **ACCESSKEY** and **SECRETKEY** correspond to a user on the S3 service. The user must have permission to perform actions on the service.

  For MinIO deployments, see Access Management for more information on user access permissions. For other S3-compatible services, defer to the documentation for that service.

Command Quick Reference

The following table lists `mc` commands:

> **Note:**
>
> The MinIO Client also includes an administration extension for managing MinIO deployments. See `mc admin` for more complete documentation.
>
> The below table does not include those commands.

Command	Description

`mc alias list`
`mc alias remove`
`mc alias set`
	The `mc alias` commands provide a convenient interface for managing the list of S3-compatible hosts that `mc` can connect to and run operations against.

`mc anonymous get`
`mc anonymous get-json`
`mc anonymous links`
`mc anonymous list`
`mc anonymous set`
`mc anonymous set-json`
	The `mc anonymous` command supports setting or removing anonymous policies to a bucket and its contents. Buckets with anonymous policies allow public access where clients can perform any action granted by the policy without authentication.

`mc batch describe`
`mc batch generate`
`mc batch list`
`mc batch start`
`mc batch status`
	The `mc batch` commands allow you to run one or more job tasks on a MinIO deployment.

`mc cat`	The `mc cat` command concatenates the contents of a file or object to another file or object. You can also use the command to display the contents of the specified file or object to `STDOUT`. `cat` has similar functionality to `cat`.

Command	Description
`mc cp`	The `mc cp` command copies objects to or from a MinIO deployment, where the source can MinIO *or* a local filesystem.
`mc diff`	The `mc diff` mc computes the differences between two filesystem directories or MinIO buckets. `mc diff` lists only those objects which are missing or which differ in size. `mc diff` does **not** compare the contents of objects.
`mc du`	The `mc du` command summarizes the disk usage of buckets and folders. You can also use `du` against the local filesystem to produce similar results as the `du` command.

`mc encrypt clear`
`mc encrypt info`
`mc encrypt set`
	The `mc encrypt` commands set, update, or disable the default bucket Server-Side Encryption (SSE) mode. MinIO automatically encrypts objects using the specified SSE mode.

`mc event add`
`mc event ls`
`mc event rm`
	The `mc event` command supports adding, removing, and listing bucket event notifications.

`mc find`	The `mc find` command supports searching for objects on a MinIO deployment. You can also use the command to search for files on a filesystem.
`mc head`	The `mc head` command displays the first `n` lines of an object, where `n` is an argument specified to the command.

`mc idp ldap add`
`mc idp ldap disable`
`mc idp ldap enable`
`mc idp ldap info`
`mc idp ldap ls`
`mc idp ldap policy`
`mc idp ldap rm`
`mc idp ldap update`
	The `mc idp ldap` commands allow you to manage configurations to 3rd party Active Directory or LDAP Identity and Access Management (IAM) integrations.

`mc idp openid add`
`mc idp openid disable`
	The `mc idp openid` commands allow you to manage configurations to 3rd party OpenID Identity and Access Management (IAM) integrations.

Command	Description

`mc idp openid info`
`mc idp openid ls`
`mc idp openid rm`
`mc idp openid update`	

`mc idp ldap policy attach`
`mc idp ldap policy detach`
`mc idp ldap policy entities`
	The `mc idp ldap policy` commands show the mapping relationships between policies and the associated groups or users.

`mc ilm restore`
`mc ilm rule add`
`mc ilm rule edit`
`mc ilm rule export`
`mc ilm rule import`
`mc ilm rule ls`
`mc ilm rule rm`
`mc ilm tier add`
`mc ilm tier check`
`mc ilm tier info`
`mc ilm tier ls`
`mc ilm tier rm`
`mc ilm tier update`
	The `mc ilm` commands manage object lifecycle management rules and tiering on a MinIO deployment.

`mc legalhold clear`
`mc legalhold info`
`mc legalhold set`
	The `mc legalhold` command sets, removes, or retrieves the Object Legal Hold (WORM) settings for object(s).

`mc license info`
`mc license register`
`mc license update`
	The `mc license` commands work with cluster registration for MinIO SUBNET. Use the commands to register a deployment, unregister a deployment, display information about the cluster’s current license, or update the license key for a cluster.

`mc ls`	The `mc ls` command lists buckets and objects on MinIO or another S3-compatible service.
`mc mb`	The `mc mb` command creates a new bucket or directory at the specified path.

Command	Description
`mc mirror`	The `mc mirror` command synchronizes content to MinIO deployment, similar to the `rsync` utility. `mc mirror` supports filesystems, MinIO deployments, and other S3-compatible hosts as the synchronization source.
`mc mv`	The `mc mv` command moves an object from source to the target, such as between MinIO deployments *or* between buckets on the same MinIO deployment. `mc mv` also supports moving objects between a local filesystem and MinIO.
`mc od`	The `mc od` command copies a local file to a remote location in a specified number of parts and part sizes. The command outputs the time it took to upload the file.
`mc ping`	The `mc ping` command performs a liveness check on a specified target.
`mc pipe`	The `mc pipe` command streams content from STDIN to a target object.

`mc quota clear`
`mc quota info`
`mc quota set`
	The `mc quota` commands configure, display, or remove a hard quota limit on a bucket.

`mc rb`	The `mc rb` command removes one or more buckets on MinIO *or* another S3-compatible service. To remove only the contents of a bucket, use `mc rm` instead.

`mc replicate add`
`mc replicate backlog`
`mc replicate export`
`mc replicate import`
`mc replicate ls`
`mc replicate resync`
`mc replicate rm`
	The `mc replicate` command configures and manages the Server-Side Bucket Replication for a MinIO deployment, including active-active replication configurations and resynchronization.

Command	Description
`mc replicate update`	

`mc retention clear`
`mc retention info`
`mc retention set`
	The `mc retention` command configures the Write-Once Read-Many (WORM) locking settings for an object or object(s) in a bucket. You can also set the default object lock settings for a bucket, where all objects without explicit object lock settings inherit the bucket default.

`mc rm`	The `mc rm` command removes objects from a bucket on a MinIO deployment. To completely remove a bucket, use `mc rb` instead.

`mc share download`
`mc share ls`
`mc share upload`
	Use the `mc share` commands to manage presigned URLs for downloading and uploading objects to a MinIO bucket.

`mc sql`	The `mc sql` command provides an S3 Select interface for performing sql queries on objects in the specified MinIO deployment.
`mc stat`	The `mc stat` command displays information on objects in a MinIO bucket, including object metadata.

`mc support callhome`
`mc support diag`
`mc support inspect`
`mc support perf`
`mc support profile`
`mc support proxy`
`mc support top api`
`mc support top disk`
`mc support top locks`
	The MinIO Client `mc support` commands provides tools for analyzing deployment health or performance and for running diagnostics. You can also upload generated health reports for further analysis by MinIO engineering.

`mc tag list`
`mc tag remove`
`mc tag set`
	The `mc tag` command adds, removes, and lists tags associated to a bucket or object.

`mc tree`	The `mc tree` command lists all prefixes inside a MinIO

Command	Description
	listing all objects inside of bucket at each prefix, including the bucket root.
`mc undo`	The `mc undo` command reverses changes due to either a `PUT` or `DELETE` operation at a specified path.
`mc update`	The `mc update` command automatically updates the **mc** binary to the latest stable version.

`mc version enable`
`mc version info`
`mc version suspend`
	The `mc version` commands enable, disable, and retrieve the versioning status for a MinIO bucket.

`mc watch`	The `mc watch` command watches for events on the specified MinIO bucket or local filesystem path. For S3 services, use `mc event add` to configure bucket event notifications on S3-compatible services.

Configuration File

`mc` uses a `JSON` formatted configuration file used for storing certain kinds of information, such as the `aliases` for each configured S3-compatible service.

For Linux and OSX, the default configuration file location is `~/.mc/config.json` .

For Windows, `mc` attempts to construct a default file path by trying specific environment variables. If a variable is unset, `mc` moves to the next variable. If all attempts fail, `mc` returns an error. The following list describes each possible file path location in the order `mc` checks them:

1. `HOME\.mc\config.json`
2. `USERPROFILE\.mc\config.json`
3. `HOMEDRIVE+HOMEPATH\.mc\config.json`

You can use the `--config-dir`

Certificates

The MinIO Client stores certificates and CAs for deployments to the following paths:

Linux, MacOS, and other Unix-like systems:

```
~/.mc/certs/CAs/ # Certificate Authorities
```

Windows systems:

```
C:\Users\[username]\mc\certs\ # certificates
C:\Users\[username]\mc\certs\CAs\ # Certificate Authorities
```

When creating a new alias, the MinIO Client fetches the peer certificate, computes the public key fingerprint, and asks the user whether to accept the deployment’s certificate. If you decide to trust the certificate, the MinIO Client adds the certificate to the certificate authority path listed above.

> **Note:**
>
> In testing environments, you can bypass the certificate check for selected MinIO Client commands by passing the `--insecure` flag.

Global Options

All commands support the following global options:

--debug

Enables verbose output to the console.

For example, the following operation adds verbose output to the `mc ls` command:

```
mc --debug ls play
```

--config-dir

The path to a `JSON` formatted configuration file that **mc** uses for storing data. See Configuration File for more information on how **mc** uses the configuration file.

--JSON

Enables JSON lines formatted output to the console.

For example, the following operation adds JSON Lines output to the `mc ls` command:

```
mc --JSON ls play
```

Disables the built-in color theme for console output. Useful for dumb terminals.

`--quiet`

Suppresses console output.

`--insecure`

Disables TLS/SSL certificate verification. Allows TLS connectivity to servers with invalid certificates. Exercise caution when using this option against untrusted S3 hosts.

`--version`

Displays the current version of `mc` .

`--help`

Optional

Displays a summary of command usage on the terminal.