

Google Cloud Storage

Paths are specified as `remote:bucket` (or `remote:` for the `lsd` command.) You may put subdirectories in too, e.g. `remote:bucket/path/to/dir`.

The initial setup for google cloud storage involves getting a token from Google Cloud Storage which you need to do in your browser.`rclone config` walks you through it.

Here is an example of how to make a remote called `remote`. First run:

```
rclone config
```

This will guide you through an interactive setup process:

```

n) New remote
d) Delete remote
q) Quit config
e/n/d/q> n
name> remote
Type of storage to configure.
Choose a number from below, or type in your own value
[snip]
XX / Google Cloud Storage (this is not Google Drive)
    \ "google cloud storage"
[snip]
Storage> google cloud storage
Google Application Client Id - leave blank normally.
client_id>
Google Application Client Secret - leave blank normally.
client_secret>
Project number optional - needed only for list/create/delete buckets - see your developer console
project_number> 12345678
Service Account Credentials JSON file path - needed only if you want use SA instead of interactive auth
service_account_file>
Access Control List for new objects.
Choose a number from below, or type in your own value
 1 / Object owner gets OWNER access, and all Authenticated Users get READER access.
    \ "authenticatedRead"
 2 / Object owner gets OWNER access, and project team owners get OWNER access.
    \ "bucketOwnerFullControl"
 3 / Object owner gets OWNER access, and project team owners get READER access.
    \ "bucketOwnerRead"
 4 / Object owner gets OWNER access [default if left blank].
    \ "private"
 5 / Object owner gets OWNER access, and project team members get access according to their roles.
    \ "projectPrivate"
 6 / Object owner gets OWNER access, and all Users get READER access.
    \ "publicRead"
object_acl> 4
Access Control List for new buckets.
Choose a number from below, or type in your own value
 1 / Project team owners get OWNER access, and all Authenticated Users get READER access.
    \ "authenticatedRead"
 2 / Project team owners get OWNER access [default if left blank].
    \ "private"
 3 / Project team members get access according to their roles.
    \ "projectPrivate"
 4 / Project team owners get OWNER access, and all Users get READER access.
    \ "publicRead"
 5 / Project team owners get OWNER access, and all Users get WRITER access.
    \ "publicReadWrite"
bucket_acl> 2
Location for the newly created buckets.
Choose a number from below, or type in your own value
 1 / Empty for default location (US).

```

```

\ ""
2 / Multi-regional location for Asia.
\ "asia"
3 / Multi-regional location for Europe.
\ "eu"
4 / Multi-regional location for United States.
\ "us"
5 / Taiwan.
\ "asia-east1"
6 / Tokyo.
\ "asia-northeast1"
7 / Singapore.
\ "asia-southeast1"
8 / Sydney.
\ "australia-southeast1"
9 / Belgium.
\ "europe-west1"
10 / London.
\ "europe-west2"
11 / Iowa.
\ "us-central1"
12 / South Carolina.
\ "us-east1"
13 / Northern Virginia.
\ "us-east4"
14 / Oregon.
\ "us-west1"
location> 12
The storage class to use when storing objects in Google Cloud Storage.
Choose a number from below, or type in your own value
1 / Default
\ ""
2 / Multi-regional storage class
\ "MULTI_REGIONAL"
3 / Regional storage class
\ "REGIONAL"
4 / Nearline storage class
\ "NEARLINE"
5 / Coldline storage class
\ "COLDLINE"
6 / Durable reduced availability storage class
\ "DURABLE_REDUCED_AVAILABILITY"
storage_class> 5
Remote config
Use auto config?
* Say Y if not sure
* Say N if you are working on a remote or headless machine or Y didn't work
y) Yes
n) No
y/n> y
If your browser doesn't open automatically go to the following link: http://127.0.0.1:53682/a

```

```
Log in and authorize rclone for access
Waiting for code...
Got code
-----
[remote]
type = google cloud storage
client_id =
client_secret =
token = {"AccessToken":"xxxx.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx-xxxxxxxxxxxxxxxxxxxxxxxxxx"}
project_number = 12345678
object_acl = private
bucket_acl = private
-----
y) Yes this is OK
e) Edit this remote
d) Delete this remote
y/e/d> y
```

Note that rclone runs a webserver on your local machine to collect the token as returned from Google if you use auto config mode. This only runs from the moment it opens your browser to the moment you get back the verification code. This is on <http://127.0.0.1:53682/> and this it may require you to unblock it temporarily if you are running a host firewall, or use manual mode.

See all the buckets in your project

Make a new bucket

List the contents of a bucket

Sync `/home/local/directory` to the remote bucket, deleting any excess files in the bucket.

Service Account support

To get credentials for Google Cloud Platform [IAM Service Accounts](#), please head to the [Service Account](#) section of the Google Developer Console. Service Accounts behave just like normal [User](#) permissions in [Google Cloud Storage ACLs](#), so you can limit their access (e.g. make them read only). After creating an account, a JSON file containing the Service Account's credentials will be downloaded onto your machines. These credentials are what rclone will use for authentication.

To use a Service Account instead of OAuth2 token flow, enter the path to your Service Account credentials at the `service_account_file` prompt and rclone won't use the browser based authentication flow. If you'd rather stuff the contents of the credentials file into the rclone config file, you can set `service_account_credentials` with the actual contents of the file instead, or set the equivalent environment variable.

Anonymous Access

For downloads of objects that permit public access you can configure rclone to use anonymous access by setting `anonymous` to `true`. With unauthorized access you can't write or create files but only read or list those buckets and objects that have public read access.

Application Default Credentials

If no other source of credentials is provided, rclone will fall back to [Application Default Credentials](#) this is useful both when you already have configured authentication for your developer account, or in production when running on a google compute host. Note that if running in docker, you may need to run additional commands on your google compute machine - [see this page](#).

Note that in the case application default credentials are used, there is no need to explicitly configure a project number.

--fast-list

This remote supports `--fast-list` which allows you to use fewer transactions in exchange for more memory. See the [rclone docs](#) for more details.

Custom upload headers

You can set custom upload headers with the `--header-upload` flag. Google Cloud Storage supports the headers as described in the [working with metadata documentation](#)

- Cache-Control
- Content-Disposition
- Content-Encoding
- Content-Language
- Content-Type
- X-Goog-Storage-Class
- X-Goog-Meta-

Eg `--header-upload "Content-Type text/potato"`

Note that the last of these is for setting custom metadata in the form `--header-upload "x-goog-meta-key: value"`

Modified time

Google google cloud storage stores md5sums natively and rclone stores modification times as metadata on the object, under the "mtime" key in RFC3339 format accurate to 1ns.

Restricted filename characters

Character	Value	Replacement
NUL	0x00	NUL
LF	0x0A	LF
CR	0x0D	CR
/	0x2F	/

Invalid UTF-8 bytes will also be [replaced](#), as they can't be used in JSON strings.

Standard Options

Here are the standard options specific to google cloud storage (Google Cloud Storage (this is not Google Drive)).

--gcs-client-id

OAuth Client Id Leave blank normally.

- Config: client_id
- Env Var: RCLONE_GCS_CLIENT_ID
- Type: string
- Default: ""

--gcs-client-secret

OAuth Client Secret Leave blank normally.

- Config: client_secret
- Env Var: RCLONE_GCS_CLIENT_SECRET
- Type: string
- Default: ""

--gcs-project-number

Project number. Optional - needed only for list/create/delete buckets - see your developer console.

- Config: project_number
- Env Var: RCLONE_GCS_PROJECT_NUMBER
- Type: string
- Default: ""

--gcs-service-account-file

Service Account Credentials JSON file path Leave blank normally. Needed only if you want use SA instead of interactive login.

Leading ~ will be expanded in the file name as will environment variables such as `${RCLONE_CONFIG_DIR}`.

- Config: service_account_file
- Env Var: RCLONE_GCS_SERVICE_ACCOUNT_FILE
- Type: string
- Default: ""

--gcs-service-account-credentials

Service Account Credentials JSON blob Leave blank normally. Needed only if you want use SA instead of interactive login.

- Config: service_account_credentials
- Env Var: RCLONE_GCS_SERVICE_ACCOUNT_CREDENTIALS
- Type: string
- Default: ""

--gcs-anonymous

Access public buckets and objects without credentials Set to 'true' if you just want to download files and don't configure credentials.

- Config: anonymous
- Env Var: RCLONE_GCS_ANONYMOUS
- Type: bool
- Default: false

--gcs-object-acl

Access Control List for new objects.

- Config: object_acl
- Env Var: RCLONE_GCS_OBJECT_ACL
- Type: string
- Default: ""
- Examples:
 - "authenticatedRead"
 - Object owner gets OWNER access, and all Authenticated Users get READER access.
 - "bucketOwnerFullControl"
 - Object owner gets OWNER access, and project team owners get OWNER access.
 - "bucketOwnerRead"
 - Object owner gets OWNER access, and project team owners get READER access.
 - "private"

- Object owner gets OWNER access [default if left blank].
- "projectPrivate"
 - Object owner gets OWNER access, and project team members get access according to their roles.
- "publicRead"
 - Object owner gets OWNER access, and all Users get READER access.

--gcs-bucket-acl

Access Control List for new buckets.

- Config: bucket_acl
- Env Var: RCLONE_GCS_BUCKET_ACL
- Type: string
- Default: ""
- Examples:
 - "authenticatedRead"
 - Project team owners get OWNER access, and all Authenticated Users get READER access.
 - "private"
 - Project team owners get OWNER access [default if left blank].
 - "projectPrivate"
 - Project team members get access according to their roles.
 - "publicRead"
 - Project team owners get OWNER access, and all Users get READER access.
 - "publicReadWrite"
 - Project team owners get OWNER access, and all Users get WRITER access.

--gcs-bucket-policy-only

Access checks should use bucket-level IAM policies.

If you want to upload objects to a bucket with Bucket Policy Only set then you will need to set this.

When it is set, rclone:

- ignores ACLs set on buckets
- ignores ACLs set on objects
- creates buckets with Bucket Policy Only set

Docs: <https://cloud.google.com/storage/docs/bucket-policy-only>.

- Config: bucket_policy_only
- Env Var: RCLONE_GCS_BUCKET_POLICY_ONLY
- Type: bool
- Default: false

--gcs-location

Location for the newly created buckets.

- Config: location

- Env Var: RCLONE_GCS_LOCATION
- Type: string
- Default: ""
- Examples:
 - ""
 - Empty for default location (US).
 - "asia"
 - Multi-regional location for Asia.
 - "eu"
 - Multi-regional location for Europe.
 - "us"
 - Multi-regional location for United States.
 - "asia-east1"
 - Taiwan.
 - "asia-east2"
 - Hong Kong.
 - "asia-northeast1"
 - Tokyo.
 - "asia-south1"
 - Mumbai.
 - "asia-southeast1"
 - Singapore.
 - "australia-southeast1"
 - Sydney.
 - "europe-north1"
 - Finland.
 - "europe-west1"
 - Belgium.
 - "europe-west2"
 - London.
 - "europe-west3"
 - Frankfurt.
 - "europe-west4"
 - Netherlands.
 - "us-central1"
 - Iowa.
 - "us-east1"
 - South Carolina.
 - "us-east4"
 - Northern Virginia.
 - "us-west1"
 - Oregon.
 - "us-west2"
 - California.

--gcs-storage-class

The storage class to use when storing objects in Google Cloud Storage.

- Config: storage_class
- Env Var: RCLONE_GCS_STORAGE_CLASS
- Type: string
- Default: ""

- Examples:
 - ""
 - Default
 - "MULTI_REGIONAL"
 - Multi-regional storage class
 - "REGIONAL"
 - Regional storage class
 - "NEARLINE"
 - Nearline storage class
 - "COLDLINE"
 - Coldline storage class
 - "ARCHIVE"
 - Archive storage class
 - "DURABLE_REDUCED_AVAILABILITY"
 - Durable reduced availability storage class

Advanced Options

Here are the advanced options specific to google cloud storage (Google Cloud Storage (this is not Google Drive)).

--gcs-token

OAuth Access Token as a JSON blob.

- Config: token
- Env Var: RCLONE_GCS_TOKEN
- Type: string
- Default: ""

--gcs-auth-url

Auth server URL. Leave blank to use the provider defaults.

- Config: auth_url
- Env Var: RCLONE_GCS_AUTH_URL
- Type: string
- Default: ""

--gcs-token-url

Token server url. Leave blank to use the provider defaults.

- Config: token_url
- Env Var: RCLONE_GCS_TOKEN_URL
- Type: string
- Default: ""

--gcs-encoding

This sets the encoding for the backend.

See: the [encoding section in the overview](#) for more info.

- Config: encoding
- Env Var: RCLONE_GCS_ENCODING
- Type: MultiEncoder
- Default: Slash,CrLf,InvalidUtf8,Dot

Limitations

`rcclone about` is not supported by the Google Cloud Storage backend. Backends without this capability cannot determine free space for an rclone mount or use policy `mfs` (most free space) as a member of an rclone union remote.