

S3 Command-Line Tool (api behavior)

This will cover the (AWS) S3 Command-Line Tool as it fits with the B2 Cloud Storage service. There are tons of commands/calls in the native S3 api that do not apply to our B2 service. Here are a list of commands and their behaviors that are relevant to our service.

- [Universal notes](#)
 - [What is a "profile"?](#)
- `aws configure`
 - [Example of profiles](#)
- `aws s3 mb`
- `aws s3 rb`
- `aws s3 ls (list buckets)`
- `aws s3 ls (list files)`
- `aws s3 cp`
- `aws s3 mv`
- `aws s3 sync`
- `aws s3 presign`

Universal notes

- All calls made within the AWS CLI will reference a "profile".
 - You can reference a specific profile by adding `--profile PROFILENAME`. Example in the `aws configure` section.
 - You can allow each call to reference the "default" profile by omitting `--profile`. Using `--profile default` will also also reference the "default" profile
- A unique authorization token will be given to each call, it is not stored in a local file somewhere and a current session is not assigned a token.
- The B2 bucket that the AWS CLI is to interact with must be created on or after May 4th, 2020. The S3 compatible API will not work on any buckets created on or before May 3rd, 2020.

What is a "profile"?

A profile is an entry in the local AWS files that the CLI will reference. Each profile will be saved in the local AWS files with the `aws configure` call and will be the pair of **key id** and **application key** (credentials).

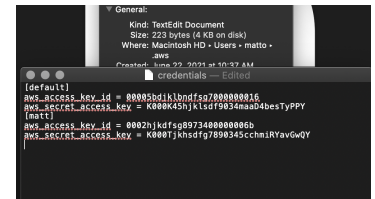
A profile's credentials are saved in a local file at:

- `~/.aws/credentials` (on Linux/Mac)
- `C:\Users\USERNAME\aws\credentials` (on Windows)

A profile's settings are saved in a local file at:

- `~/.aws/config` (on Linux/Mac)
- `C:\Users\USERNAME\aws\config` (on Windows)

If either of the above don't yet exist, they will be created when `aws configure` is called.



[Link to screenshot](#)

These keys are fake and will not work.



Apart from `aws configure`, in order to use the S3 compatible API to access B2 Cloud Storage, all calls must have `--endpoint-url=https://<YOUR S3 ENDPOINT URL>`. This `<S3 ENDPOINT URL>` can be found on the Buckets page of your account after signing into https://secure.backblaze.com/user_signin.htm.

The S3 endpoint on the Buckets page will not have `https://` and this must be added when using the AWS CLI. Most integrations will add this by default.

The S3 endpoint can go anywhere in the call after `aws`. For the sake of consistency in this guide, it will only ever go at the end.

aws configure

Compare to `b2 authorize-account`



`aws configure` [`--profile PROFILE`]

Required information:

- Your B2 keyId and applicationKey
 - Must be a created applicationKey. The Master will not work
 - If the applicationKey is restricted to a specific bucket, it must have the permission called "listAllBucketNames".
 - When the AWS CLI asks for the "AWS Access Key ID", we will enter the **keyId**. For the "AWS Secret Access Key" we will enter the **applicationKey**.

Notes:

This call doesn't request an authorization token. This call is used to add a "profile" to your AWS CLI as well as update the configuration settings for the specified "profile".

When making this call, the CLI will ask you to input a "Default region name" and then a "Default output format". Leave these blank, it will break some of the calls if this is present; you can just press enter to continue past this prompt.

Optional arguments:

You can run `aws configure` (with or without `--profile`) and successfully add non-credentials to a profile. Non-credentials being strings of characters that will not successfully authorize the usage of an account (B2 or S3 account).

If `aws configure` is ran without `--profile`, it will still prompt you for credentials. When you complete the prompts, it will set your entries to a profile called "default". If the "default" profile already exists, you can either enter new credentials or press enter/return to use the credentials for the default profile.

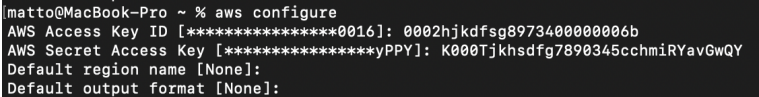
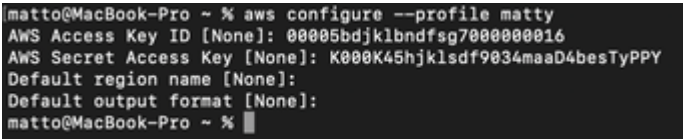
Fixing a profile:

If a mistake is made when creating a profile, you can run `aws configure` one more time and it will overwrite the profile. If you need to remove something that should not be there, you can just type in "None" and it will change it to the default settings.

Alternatively, you can just delete the local files as described in the "What is a profile?" section above.

Output:

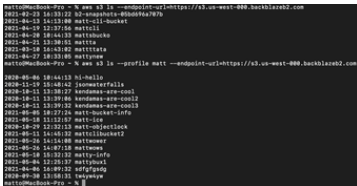
You won't see any response from the CLI after running this call completely. You'll only see the CLI prompting you to enter your credentials.

Input	Output
<p>aws configure</p> <p>Without an existing default profile.</p> <p>Since this call does not have <code>--profile</code> it will be referencing the "default" profile. Since there is no existing default profile, you see [None] show up for both keyId and applicationKey. This will create the default profile.</p>	 <p>matto@MacBook-Pro ~ % aws configure AWS Access Key ID [None]: 00005bdjklbndfsg7000000016 AWS Secret Access Key [None]: K000K45hjkl sdf9034maaD4besTyPPY Default region name [None]: Default output format [None]: matto@MacBook-Pro ~ %</p> <p>Link to screenshot</p> <p>Fake keys.</p>
<p>aws configure</p> <p>With an existing default profile.</p> <p>Since this call also does not have <code>--profile</code>, it will again reference the "default" profile. Since there is an existing default profile, it will show the last 4 digits of the keyId and applicationKey. This will overwrite the current default profile.</p>	 <p>matto@MacBook-Pro ~ % aws configure AWS Access Key ID [*****0016]: 0002hjdkfsg8973400000006b AWS Secret Access Key [*****yPPY]: K000Tjkh sdfg7890345cchmiRYavGwQY Default region name [None]: Default output format [None]:</p> <p>Link to screenshot</p> <p>Fake keys.</p>
<p>aws configure <code>--profile matt</code></p> <p>Without an existing profile called matt. This call will just create a new profile called matt.</p> <p>This will have the same overwriting behavior as the call above.</p>	 <p>matto@MacBook-Pro ~ % aws configure --profile matt AWS Access Key ID [None]: 00005bdjklbndfsg7000000016 AWS Secret Access Key [None]: K000K45hjkl sdf9034maaD4besTyPPY Default region name [None]: Default output format [None]: matto@MacBook-Pro ~ %</p> <p>Link to screenshot</p> <p>Fake keys.</p>

Example of profiles

Currently, I have 2 profiles in my AWS CLI, the default profile and a profile called “matt”. I am able to list my buckets for the default profile and then immediately make the same call but reference the “matt” profile.

You do not need to re-authenticate with another set of credentials to list buckets for another profile.



[Link to screenshot](#)

aws s3 mb

Compare to b2 create-bucket

```
aws s3 mb s3://<BUCKET NAME> --region <YOUR REGION> [--profile PROFILENAME] --endpoint-url=https://<YOUR S3 ENDPOINT URL>
```

Required information:

- Your S3 endpoint url.
 - --region must be included and can be taken from part of the endpoint url.
- User chosen bucket name.
 - Must begin with s3://
 - Must follow the normal bucket naming guidelines. <https://www.backblaze.com/b2/docs/buckets.html>

Notes:

Makes a B2 bucket.

This call will requires the --region entry in addition to the --endpoint-url. The region can be taken from your S3 Endpoint.

The bucket this creates will be **Private** by default and also with default settings.

Output:

Input	Output
aws s3 mb s3://anothermattbucket --region us-west-000 --endpoint-url=https://s3.us-west-000.backblazeb2.com	<pre>matto@MacBook-Pro ~ % aws s3 mb s3://anothermattbucket --region us-west-000 --endpoint-url=https://s3.us-west-000.backblazeb2.com make_bucket: anothermattbucket</pre> <p>Link to screenshot</p>
aws s3 mb s3://anothermattbucket --endpoint-url=https://s3.us-west-000.backblazeb2.com  This call fails because the --region was not included.	<pre>s3.us-west-000.backblazeb2.com make_bucket failed: s3://anothermattbucket An error occurred (InvalidArgument) when calling the CreateBucket operation: Location constraint specifies another region, Backblaze does not support cross-region requests</pre> <p>Link to screenshot</p>

aws s3 rb

Compare to b2 delete-bucket

```
aws s3 rb s3://<BUCKET NAME> [--profile PROFILENAME] --endpoint-url=https://<YOUR S3 ENDPOINT URL>
```

Required information:

- Your S3 endpoint.
- Name of bucket that will be deleted.

Notes:

This will remove the specified bucket.

The bucket must be empty for this call to be successful. It will not delete a bucket with objects aside from .DS_STORE or empty “folders”.

Optional arguments:


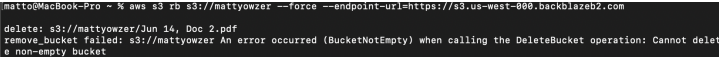
- ❌ The AWS CLI documentation mentions the optional `[--force]`. This option will delete all **non-versioned** objects in a bucket and then delete the bucket if it is empty. However, the call will treat all objects in a B2 bucket as **versioned** files and will only hide whatever objects are in the bucket. The call will then fail to delete the bucket because it is not empty.

Solution

You can use the B2 CLI API to delete everything in the bucket and then delete the bucket. To so do, run `b2 sync --delete --allowEmptySource <BUCKET NAME> <Local path to an empty folder>` to delete everything in the bucket and then use `b2 delete-bucket <BUCKET NAME>` or `aws s3 rb` to delete the bucket. More information on that here <https://help.backblaze.com/hc/en-us/articles/225556127-How-Can-I-Easily-Delete-All-Files-in-a-Bucket->.


The above `b2 sync` command will sometimes upload a `“.DS_STORE”` file (on a Mac) which should be deleted in order to successfully delete the bucket.

Output:

Input	Output
<pre>aws s3 rb s3://mattsbucko --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre>	 <pre>matto@MacBook-Pro ~ % aws s3 rb s3://mattsbucko --endpoint-url=https://s3.us-west-000.backblazeb2.com remove_bucket: mattsbucko</pre> <p>Link to screenshot</p>
<p>❌ <pre>aws s3 rb s3://mattyowzer --force --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre></p> <p>This call fails because there was an object in the bucket. <code>--force</code> makes the call say that the object was deleted, however, the object was actually only hidden and the call fails because the bucket is not empty.</p>	 <pre>matto@MacBook-Pro ~ % aws s3 rb s3://mattyowzer --force --endpoint-url=https://s3.us-west-000.backblazeb2.com delete: s3://mattyowzer/Jun 14, Doc 2.pdf remove_bucket failed: s3://mattyowzer An error occurred (BucketNotEmpty) when calling the DeleteBucket operation: Cannot delete a non-empty bucket</pre> <p>Link to screenshot</p>

aws s3 ls (list buckets)

Compare to `b2 list-buckets`.

 `aws s3 ls [--profile PROFILENAME] --endpoint-url=https://<YOUR S3 ENDPOINT URL>`

Required information:

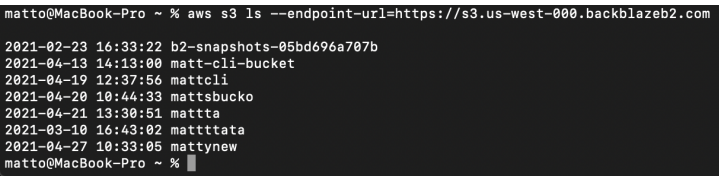
- Your S3 endpoint.

Notes:

This will list items at the given location.

By specifying only the endpoint `--endpoint-url=https://<YOUR S3 ENDPOINT URL>`, the location the call will be referencing is the Backblaze account. This will list all buckets in the Backblaze account.

Output:

Input	Output
<pre>aws s3 ls --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre> <p>Lists only buckets in the default profile.</p>	 <pre>matto@MacBook-Pro ~ % aws s3 ls --endpoint-url=https://s3.us-west-000.backblazeb2.com 2021-02-23 16:33:22 b2-snapshots-05bd696a707b 2021-04-13 14:13:00 matt-cli-bucket 2021-04-19 12:37:56 mattcli 2021-04-20 10:44:33 mattsbucko 2021-04-21 13:30:51 mattta 2021-03-10 16:43:02 mattttata 2021-04-27 10:33:05 mattynew matto@MacBook-Pro ~ %</pre> <p>Link to screenshot</p>

aws s3 ls (list files)

Compare to `b2 ls`.

```
aws s3 ls [s3://<BUCKET NAME>/<PATH>] [--human-readable] [--recursive] [--summarize] [--profile PROFILENAME] --endpoint-url=https://<YOUR S3 ENDPOINT URL>
```

Required information:

- Your S3 endpoint url.
- Since we will be listing files with this call, you will also need the bucket name and/or bucket path.

Notes:

This will list objects at the given location.

By specifying the bucket name/path [s3://<BUCKET NAME>/<PATH>], the location will reference a specific bucket and/or a nested path within it. This will list all objects in a bucket or path.

- To list objects in a specific folder in a B2 bucket, the path needs the trailing backslash.
- If making the call without the trailing backslash, the output will only list all folders that begin with the last part of the path.


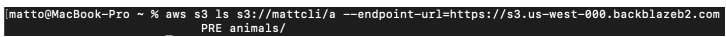

Optional arguments:



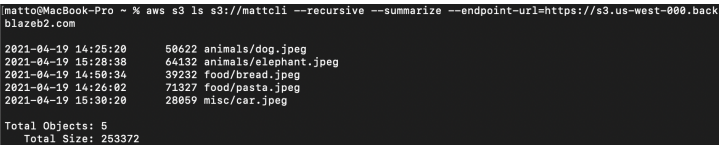
- Adding [--recursive] in the call with the bucket path, will allow the call to iterate through items nested in subfolders.
- By default, when listing files, the call will display the size of each file in bytes aka base 1024. Adding [--human-readable] will make the call display the size of each file in KiB aka Kilobytes in base 1000. This will not have an effect on listing buckets.
- Adding [--summarize] will make the call summarize the **total number of files** and the **total size of all files listed** in bytes only if they are listed from the call. If only folders or only buckets are listed, they will not be add to the total number of objects or size.
 - [--human-readable] will make the **total size of all files listed** displayed in KiB rather than bytes.

▼ This is the structure of the folder used for the following examples.

- mattcli
 - animals
 - dog.jpeg
 - elephant.jpeg
 - food
 - bread.jpeg
 - pasta.jpeg
 - misc
 - car.jpeg

Output:

Input	Output
<pre>aws s3 ls s3://mattcli --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre> <p>Lists only the top level items. This example is folders only but files will also be listed.</p>	 <p>Link to screenshot</p>
<pre>aws s3 ls s3://mattcli/a --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre> <p>Interestingly, adding a partial folder/file name for a bucket path will tell the call to list anything in the top level that matches that partial name.</p>	 <p>Link to screenshot</p>
<pre>aws s3 ls s3://mattcli/food --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre> <p>Without the trailing backslash on the bucket path will only list folders that begin with the last part of the path.</p> <pre>aws s3 ls s3://mattcli/food/ --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre> <p>With the trailing backslash on the bucket path will list all objects within the folder that matches the last part of the path.</p>	 <p>Link to screenshot</p>

<pre>aws s3 ls s3://mattcli --recursive --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre> <p>--recursive lists all items including nested files/folders.</p> <p>This will also work with partial folder/file names as shown in the example above.</p>	 <p>Link to screenshot</p>
<pre>aws s3 ls s3://mattcli --recursive --human-readable --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre> <p>--human-readable changes the file size in bytes to KiB.</p>	 <p>Link to screenshot</p>
<pre>aws s3 ls s3://mattcli --recursive --summarize --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre>	 <p>Link to screenshot</p>

aws s3 cp

Compare to b2 upload-file, b2 copy-file-by-id and b2 download-file-by-id/b2 download-file-by-name

i `aws s3 cp source destination [--dryrun] [--quiet] [--content-type CONTENTTYPE] [--sse AES256] [--profile PROFILENAME] --endpoint-url=https://<YOUR S3 ENDPOINT URL>`

Required information:

- Your S3 endpoint url.
- Source path with file name.
- Destination path with file name.

Notes:

This is the same as `aws s3 mv` but will not delete the object from the source path.

If the destination is a local or bucket path that does not exist, it will create new folders when the call is made.

✓ This can be used to do the following things:

Upload a file from a local path to a bucket path.	source destination <Local path> <Bucket path>
Copy a file from a bucket to another bucket path in the same account. This happens server-side.	source destination <Bucket path> <Bucket path>
Download a file from a bucket path to a local path.	source destination <Bucket path> <Local path>

✗ This cannot be used to move a file from a local path to another local path.

Local and bucket path rules

- Local and bucket paths require a trailing file name if it is the source.
- A local path does not require a trailing file name if it is the destination.
- A bucket path does require the trailing file name if it is the destination.

If a destination bucket path doesn't have a trailing file name, the call will still respond as if it were successful but nothing will be uploaded or copied server-side.

Optional arguments:

`--dryrun` will simulate the upload/download/copy process without actually carrying the command out.

`--quiet` will carry the command out with no response from the API.




`--content-type CONTENTTYPE` will set the content-type for the file uploaded. If this is left out of the upload call, it will default according to the file's extension. Here is a list of possible content-types: <https://www.backblaze.com/b2/docs/content-types.html>

`--sse AES256` sets the file being uploaded or copied (not downloaded) to be encrypted at rest using Advanced Encryption Standard (AES-256). Encrypted objects in a bucket cannot be downloaded through the Web UI and cannot be included in a snapshot; they must be downloaded through the API.

`--sse-c` and `--sse-c-key` are not supported in the AWS CLI (S3 compatible API).

Output:

This will give back a message stating what the API did.

Input	Output
<pre>aws s3 cp /Users/matto/Documents/files/food/car.jpeg s3://mattcli/test2/car2.jpeg --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre> <p>To upload a file: source destination <Local path> <Bucket path></p>	 <p>Link to screenshot</p>
<pre>aws s3 cp s3://mattcli/food/bread.jpeg s3://mattcli/hi/bread.jpeg --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre> <p>To copy a file: source destination <Bucket path> <Bucket path></p>	 <p>Link to screenshot</p>
<pre>aws s3 cp s3://mattcli/food/bread.jpeg /Users/matto/Desktop/files/2/animals --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre> <p>To download a file: source destination <Bucket path> <Local path></p>	 <p>Link to screenshot</p>
<p>✗ <pre>aws s3 cp /Users/matto/Documents/files/food s3://mattcli/test2 --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre></p> <p>This fails because the source local path doesn't have a trailing file name.</p> <p>This will also fail if the destination location doesn't have a file name.</p>	 <p>Link to screenshot</p>
<p>✗ <pre>aws s3 cp /Users/matto/Documents/files/food/car.jpeg s3://mattcli/test2 --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre></p> <p>The output looks successful (it fails), but since the destination bucket path doesn't have a trailing file name, nothing will actually be uploaded.</p>	 <p>Link to screenshot</p>

aws s3 mv

Compare to `b2 upload-file`, `b2 copy-file-by-id` and `b2 download-file-by-id`/`b2 download-file-by-name` **PLUS** `b2 delete-file-version`

 `aws s3 mv source destination [--profile PROFILENAME] --endpoint-url=https://<YOUR S3 ENDPOINT URL>`

Required information:

- Your S3 endpoint url.
- Source path with file name.
- Destination path with file name.

Notes:

This is the same as `aws s3 cp` and will delete the object from the `source` path.

If the `destination` is a local or bucket path that does not exist, it will create new folders when the call is made.

✓ This can be used to do the following things:

Upload a file from a local path to a bucket path....and then delete the <code>source</code> object.	<code>source destination <Local path> <Bucket path></code>
Copy a file from a bucket to another bucket path in the same account....and then delete the <code>source</code> object. This happens server-side.	<code>source destination <Bucket path> <Bucket path></code>
Download a file from a bucket path to a local path....and then delete the <code>source</code> object.	<code>source destination <Bucket path> <Local path></code>

✗ This cannot be used to move a file from a local path to another local path.

Local and bucket path rules:

- Local and bucket paths require a trailing file name if it is the `source`.
- Local and bucket paths do not require a trailing file name if it is the `destination`.

Optional arguments:

`--dryrun` will simulate the upload/download/copy process without actually carrying the command out.

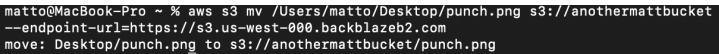
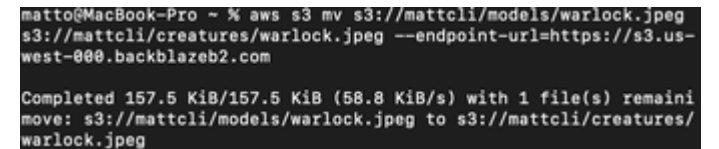

`--quiet` will carry the command out with no response from the API.

`--content-type CONTENTTYPE` will set the content-type for the file uploaded. If this is left out of the upload call, it will default according to the file's extension. Here is a list of possible content-types: <https://www.backblaze.com/b2/docs/content-types.html>

`--sse AES256` sets the file being uploaded or copied (not downloaded) to be encrypted at rest using Advanced Encryption Standard (AES-256). Encrypted objects in a bucket cannot be downloaded through the Web UI and cannot be included in a snapshot; they must be downloaded through the API.

`--sse-c` and `--sse-c-key` are not supported in the AWS CLI (S3 compatible API).

Output:

Input	Output
<pre>aws s3 mv /Users/matto/Desktop/punch.png s3://another mattbucket --endpoint-url=https://s3.us-west-000. backblazeb2.com</pre> <p>To upload a file: <code>source destination <Local path> <Bucket path></code></p> <p>...and then delete the <code>source</code> object.</p>	 <p>Link to screenshot</p>
<pre>aws s3 mv s3://mattcli/models/warlock.jpeg s3://mattc li/creatures/warlock.jpeg --endpoint- url=https://s3.us-west-000.backblazeb2.com</pre> <p>To copy a file: <code>source destination <Bucket path> <Bucket path></code></p> <p>...and then delete the <code>source</code> object.</p>	 <p>Link to screenshot</p>
<pre>aws s3 mv s3://anothermattbucket/punch.png /Users /matto/Desktop/files2 --endpoint-url=https://s3. us-west-000.backblazeb2.com</pre> <p>To download a file: <code>source destination <Bucket path> <Local path></code></p> <p>...and then delete the <code>source</code> object.</p>	 <p>Link to screenshot</p>

aws s3 sync

Compare to `b2 sync`

```
aws s3 sync source destination [--dryrun] [--quiet] [--sse AES256] --endpoint-url=https://<YOUR S3 ENDPOINT URL>
```

Required information:

- Your S3 endpoint url.
- Source folder path.
- Destination folder path.

Notes:

If the `destination` is a local folder path or bucket folder path that does not exist, it will create new folders when the call is made.

✓ This can be used to do the following things:

- Sync a local folder path to a bucket folder path. This will upload from your computer to a bucket.
 - `source destination <Local folder path> <Bucket folder path>`
- Sync a bucket to another bucket path in the same account. This will copy contents of a bucket folder path to another bucket folder path. This can be within the same bucket or to another bucket in the same account and will happen server-side.
 - `source destination <Bucket folder path> <Bucket folder path>`
- Sync a bucket to a local path. This will download from a bucket to your computer.
 - `source destination <Bucket folder path> <Local folder path>`

✗ This cannot be used to sync a local folder path to another local folder path.

✗ This will only work on two different folder paths. A specific trailing file name cannot be included in the path or the call will fail.

Optional arguments:

`--dryrun` will simulate the upload/download/copy process without actually carrying the command out. Each object upload that was demonstrated will say (dryrun) next to it.

`--quiet` will carry the command out with no response from the API.

`--sse AES256` sets files being uploaded or copied (not downloaded) to be encrypted at rest using Advanced Encryption Standard (AES-256). Encrypted objects in a bucket cannot be downloaded through the Web UI and cannot be included in a snapshot; they must be downloaded through the API.

`--sse-c` and `--sse-c-key` are not supported in the AWS CLI (S3 compatible API)

`--delete` will hide objects in a bucket if it is a `destination` that aren't at the `source`. It will delete objects that are non-version, but this call will treat all objects in a bucket as versioned files and will only hide them rather than deleting them.

Exclusions and inclusions:

Currently, there is no support for the use of UNIX style wildcards in a command's path arguments. However, most commands have `--exclude <value>` and `--include <value>` parameters that can achieve the desired result. These parameters perform pattern matching to either exclude or include a particular file or object. The following pattern symbols are supported.

"": Matches everything*

"?": Matches any single character

[sequence]: Matches any character in

sequence[!sequence]: Matches any character not in sequence "


Any number of these parameters can be passed to a command. You can do this by providing an `--exclude` or `--include` argument multiple times, e. g. `--include "*.txt" --include "*.png"`. When there are multiple filters, the rule is the filters that appear later in the command take precedence over filters that appear earlier in the command. For example, if the filter parameters passed to the command were

```
--exclude "*" --include "*.txt"
```

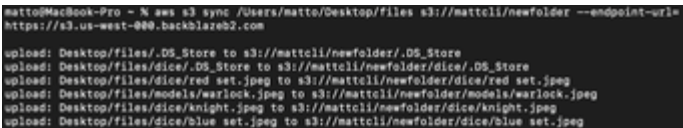



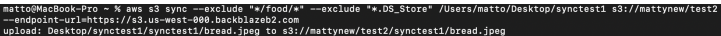
All files will be excluded from the command except for files ending with `.txt` However, if the order of the filter parameters was changed to

```
--include "*.txt" --exclude "*" 
```

All files will be excluded from the command.

 Pathing format differences between Windows and Mac must be respected.

Output:

Input	Output
<pre>aws s3 sync /Users/matto/Desktop/files s3://mattcli/newfolder --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre> <p>To sync a local folder path to a bucket folder path (upload).</p>	 <p>Link to screenshot</p>
<pre>aws s3 sync s3://mattcli/sync1 s3://mattcli/sync2 --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre> <p>To sync a bucket folder path to another bucket folder path (server-side copy)</p>	 <p>Link to screenshot</p>
<pre>aws s3 sync s3://mattcli/newfolder /Users/matto/Desktop/files2/folder --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre> <p>To sync a bucket folder path to a local folder path (download).</p>	 <p>Link to screenshot</p>
<p>Exclusions</p> <pre>aws s3 sync /Users/matto/Desktop/synctest1 s3://mattynew/test1 --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre> <p>This is what it looks like when syncing a folder normally.</p>	 <p>Link to screenshot</p>
<p>Exclusions (cont.)</p> <pre>aws s3 sync --exclude "**/food/" --exclude "**.DS_Store" /Users/matto/Desktop/synctest1 s3://mattynew/test2 --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre> <p>Syncing the same folder as above but:</p> <ul style="list-style-type: none">Excludes all folders called <code>food</code>Excludes all files called <code>.DS_Store</code>	 <p>Link to screenshot</p>

```
aws s3 sync --exclude "*/food/*" --exclude "*/.DS_Store" --include "*/
jpeg" /Users/matto/Desktop/sync_test1 s3://mattynew
/test3 --endpoint-url=https://s3.us-west-000.
backblazeb2.com
```

- Still excludes all folders called *food*
- Still excludes all files called *.DS_Store*
- Includes all *.jpeg* files.
 - This will include all of those files even if they match an excluded phrase.

[Link to screenshot](#)

[Link to screenshot](#)

Input	Output
<pre>aws s3 presign s3://mattsbucko/file1.png --endpoint-url=https://s3.us-west-000.backblazeb2.com</pre>	<pre>matt@MacBook-Pro ~ % aws s3 presign s3://mattsbucko/file1.png --endpoint-url=https://s3.us-west-000.backblazeb2.com https://s3.us-west-000.backblazeb2.com/mattsbucko/file1.png?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-Credential=00808bd96a7076b0e0800c4d0f2921e61cd4Fu--mat-s3v1sgFw4s+request&X-Amz-Date=20230316T173530Z&X-Amz-Expires=3600&X-Amz-SignedHeaders=host&X-Aws-Signature=w0r224dc3bae89e51444dd2bc4c5280290ce5c27a93dbd3be6523cab516601</pre>

[Link to screenshot](#)