s3cmd S3 command line

CONCEPTS

s3cmd reads ~/.s3cfg. For S3-compatible stores (Ceph, MinIO, Wasabi) set host_base and host_bucket there, or pass --host and --host-bucket.

~/.s3cfg keys, endpoint
↓ loads
s3cmd command + options
↓ signs and sends
Bucket s3://bucket
↓ holds
Objects key = prefix/name

A trailing slash matters: sync <dir>/ copies the contents, sync <dir> puts the directory name into the key. Always try -n first.

SETUP

s3cmd --configure
Interactive setup, writes ~/.s3cfg

-c *<file>*
Use another config file

--host=*<host>* **--host-bucket=***<tmpl>*
Point at an S3-compatible endpoint

BUCKETS

s3cmd mb s3://*<bucket>*
Make a bucket

BUCKETS (cont.)

s3cmd mb s3://*<bucket>* **--region=***<region>*
Make a bucket in a region

s3cmd rb s3://*<bucket>*
Remove an empty bucket

s3cmd ls
List your buckets

s3cmd du -H [s3://*<bucket>***]**
Disk usage, human readable

LIST OBJECTS

s3cmd ls s3://*<bucket>***/***<prefix>***/**
List one prefix

s3cmd ls -r s3://*<bucket>*
List recursively

UPLOAD AND DOWNLOAD

s3cmd put *<file>*... **s3://***<bucket>***/[***<prefix>***]**
Upload files

put -r *<dir>* **s3://***<bucket>***/**
Upload a directory tree

put -P *<file>* **s3://***<bucket>***/**
Upload world-readable

put --storage-class=*<class>*
Choose a storage class

UPLOAD AND DOWNLOAD (cont.)

put --add-header=*<Name:Value>*
Set an HTTP header such as Cache-Control

put --multipart-chunk-size-mb=*<n>*
Set the multipart chunk size

s3cmd get s3://*<bucket>***/***<key>* **[***<file>***]**
Download an object

get -r s3://*<bucket>***/***<prefix>***/** *<dir>***/**
Download a prefix

get --continue
Resume a partial download

s3cmd cp s3://*<b1>***/***<k1>* **s3://***<b2>***/***<k2>*
Copy between keys or buckets

s3cmd mv s3://*<b1>***/***<k1>* **s3://***<b2>***/***<k2>*
Move an object

SYNC

s3cmd sync *<dir>***/ s3://***<bucket>***/***<prefix>***/**
Upload changed files

s3cmd sync s3://*<bucket>***/***<prefix>***/** *<dir>***/**
Download changed objects

SYNC (cont.)

-n
Dry run: show what would happen

--delete-removed
Delete destination extras

--max-delete=*<n>*
Refuse to delete more than n

--exclude=*<glob>* **/ --include=***<glob>*
Filter by pattern

--no-check-md5
Compare sizes only (faster)

DELETE

s3cmd del s3://*<bucket>***/***<key>*
Delete an object

s3cmd rm -r s3://*<bucket>***/***<prefix>***/**
Delete a whole prefix

del --max-delete=*<n>*
Limit how many are deleted

ACCESS AND SHARING

s3cmd setacl --acl-public s3://*<bucket>***/***<key>*
Make an object public

setacl --acl-private
Restrict to the owner

ACCESS AND SHARING (cont.)

setacl -r
Apply to every object under a prefix

s3cmd signurl s3://*<bucket>***/***<key>* **+***<seconds>*
Pre-signed URL, relative expiry

DEBUG

-d
Debug output with HTTP traffic

-v
Verbose output

RECIPES

Safe backup mirror

```
s3cmd sync -n --delete-removed --max-delete=20 ./data/ s3://backups/data/
# looks right? repeat without -n
s3cmd sync --delete-removed --max-delete=20 ./data/ s3://backups/data/
```

Publish a static site

```
s3cmd sync -P --exclude='*.tmp' ./public/ s3://my-site/
```

Share a file for an hour

```
s3cmd put report.pdf s3://files/report.pdf
s3cmd signurl s3://files/report.pdf +3600
```