



AWS S3

Verified 2026-10-23

Object storage — the flat key model, CLI, storage classes, bucket policies, presigned URLs, versioning, lifecycle, and events.

Contents

Mental model	3
Buckets, keys, and limits	3
CLI essentials	4
Storage classes	5
Access control	7
Encryption	9
Presigned URLs	10
Versioning and deletion	11
Large uploads	13
Lifecycle rules	14
Events and replication	15

Further reading	16
-----------------------	----

Mental model

S3 is a flat key-value store for whole objects, not a file system. A bucket holds objects, each addressed by a key such as `logs/2026/app.gz`; the slashes are just characters, and “folders” are a console and CLI convention over key prefixes. You write or replace an object as a unit, you never edit it in place, and every request is an authenticated HTTPS call.

Buckets, keys, and limits

Item	Rule
Bucket name	3–63 chars, lowercase, globally unique
Object size	0 B to 5 TB
Single PUT	Up to 5 GB
Key length	Up to 1,024 bytes
Request rate	3,500 writes, 5,500 reads per second per prefix

A bucket lives in one Region, but its name is unique across every account in the partition. Reads after a write

are strongly consistent: a successful PUT is visible to the next GET or LIST. Scale request rate by spreading keys over more prefixes.

```
s3://my-bucket/logs/2026/app.gz
```

bucket key (prefix = logs/2026/)

Gotcha: Bucket names are public and global. A deleted bucket's name can be claimed by anyone, so a dangling reference to it is a takeover risk.

CLI essentials

Command	Does
<code>aws s3 ls</code>	List buckets or a prefix
<code>aws s3 cp</code>	Copy one object or file
<code>aws s3 sync</code>	Copy only new or changed files
<code>aws s3 rm</code>	Delete objects
<code>aws s3api</code>	Low-level, one call per API operation

`aws s3` is the high-level layer: it paginates, parallelizes, and switches to multipart for large files. Drop to `aws s3api` for anything it lacks, such as versions, tagging, or policies.

```
aws s3 mb s3://my-bucket --region eu-west-1
aws s3 cp app.log s3://my-bucket/logs/
aws s3 cp s3://my-bucket/logs/ . --recursive
aws s3 sync ./dist s3://my-bucket/site --delete
aws s3 ls s3://my-bucket/logs/ --human-readable
aws s3 rm s3://my-bucket/logs/ --recursive
```

Warning: `sync --delete` removes every object under the destination that is missing locally. Run it with `--dryrun` first.

Storage classes

Class	Use for	Min duration
STANDARD	Frequent access	None

Class	Use for	Min duration
INTELLIGENT_TIERING	Unknown or shifting access	None
STANDARD_IA	Infrequent, needs instant reads	30 days
GLACIER_IR	Archive, instant reads	90 days
GLACIER	Archive, minutes to hours	90 days
DEEP_ARCHIVE	Long-term, up to 12+ hours	180 days

You choose the class per object at upload, or let lifecycle rules move it later. Colder classes cost less to store and more to read. `ONEZONE_IA` is the cheaper single-AZ variant of `STANDARD_IA` ; its data is lost if that AZ is.

```
aws s3 cp backup.tar s3://my-bucket/ \
  --storage-class GLACIER_IR
```

Gotcha: GLACIER and DEEP_ARCHIVE objects cannot be read until you restore them. A plain GET fails with `InvalidObjectState`; call `aws s3api restore-object` and wait.

Access control

Mechanism	Attached to	Use for
IAM policy	A user or role	What a principal may do
Bucket policy	A bucket	Cross-account, conditions, public rules
Block Public Access	Account or bucket	A hard stop on public exposure
ACLs	Bucket or object	Legacy only; disabled by default

A request succeeds when some policy allows it and none denies it. New buckets have Block Public Access on and ACLs disabled (Object Ownership set to bucket owner enforced), so the bucket owner owns every object. Keep it that way and grant access with policies.

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Principal": {
      "AWS": "arn:aws:iam::111122223333:root"
    },
    "Action": "s3:GetObject",
    "Resource": "arn:aws:s3:::my-bucket/*"
  }]
}
```

Gotcha: Bucket and object actions use different resource ARNs. `s3:ListBucket` needs `arn:aws:s3:::my-bucket`; `s3:GetObject` needs `arn:aws:s3:::my-bucket/*`. Mixing them up returns `AccessDenied`.

Encryption

Option	Key held by
SSE-S3	S3 (default for all new objects)
SSE-KMS	AWS KMS key, with audit trail
SSE-C	You, sent on every request

Every object is encrypted at rest, and new objects use SSE-S3 unless you choose otherwise. Pick SSE-KMS when you need key policies or CloudTrail records of key use. Traffic is protected in transit only if you require it; deny plain HTTP with a bucket policy.

```
{  
  "Effect": "Deny",  
  "Principal": "*",  
  "Action": "s3:*",  
  "Resource": ["arn:aws:s3:::my-bucket/*"],  
  "Condition": {"Bool": {  
    "aws:SecureTransport": "false"  
  }}  
}
```

Note: With SSE-KMS, callers need `kms:Decrypt` or `kms:GenerateDataKey` on the key as well as S3 permissions.

Presigned URLs

Part	Detail
Signed by	The credentials that created it
Grants	One operation on one key
CLI default expiry	3,600 s (1 hour)

Part	Detail
Maximum expiry	7 days (IAM user keys)

A presigned URL lets someone with no AWS credentials GET or PUT a single object until it expires. It acts with the signer's permissions, so it only works while the signer can still do that action.

```
aws s3 presign s3://my-bucket/report.pdf \
  --expires-in 600
```

Gotcha: URLs signed with temporary credentials (an assumed role, Lambda, EC2) die when those credentials expire, even if you asked for a longer lifetime.

Versioning and deletion

State	Effect of DELETE key
Unversioned	Object is gone permanently

State	Effect of DELETE key
Enabled	Adds a delete marker; versions stay
Suspended	Adds a null-version delete marker
DELETE key?versionId=	Removes that version for good

With versioning on, an overwrite keeps the old data as a noncurrent version and you pay to store it. Once enabled, versioning can only be suspended, never removed. **MFA delete** and **Object Lock** protect versions from deletion.

```
aws s3api put-bucket-versioning \
  --bucket my-bucket \
  --versioning-configuration Status=Enabled
aws s3api list-object-versions --bucket my-bucket
```

Gotcha: After a delete, `ls` shows the object gone, yet storage keeps billing. Remove the delete marker to restore it, and expire noncurrent versions with lifecycle rules.

Large uploads

Fact	Value
Part size	5 MiB to 5 GiB (last part smaller)
Parts per upload	Up to 10,000
Recommended from	About 100 MB
Required above	5 GB

Multipart upload splits an object into parts that upload in parallel and retry independently. `aws s3 cp` and the SDK transfer managers do this for you above a size threshold. The object appears only after the upload is completed.

Gotcha: Abandoned multipart uploads are invisible in `ls` but still billed. Add a lifecycle rule with `AbortIncompleteMultipartUpload` after a few days.

Lifecycle rules

Action	Effect
Transition	Move to another storage class
Expiration	Delete current objects
NoncurrentVersionExpiration	Delete old versions
AbortIncompleteMultipartUpload	Clean up failed uploads

A rule applies to a prefix, tag, or whole bucket and runs asynchronously, so effects can lag by a day. Transitions must move to a colder class and respect minimum durations.

```
aws s3api put-bucket-lifecycle-configuration \  
  --bucket my-bucket \  
  --lifecycle-configuration file:///rules.json
```

```
{"Rules": [{  
  "ID": "archive-logs",  
  "Status": "Enabled",  
  "Filter": {"Prefix": "logs/"},  
  "Transitions": [  
    {"Days": 30, "StorageClass": "GLACIER_IR"}],  
  "Expiration": {"Days": 365}  
}]}
```

Events and replication

Feature	Purpose
Event notifications	Push <code>ObjectCreated</code> and similar events
EventBridge	Same events, richer filtering and routing

Feature	Purpose
Replication (CRR/SRR)	Copy new objects to another bucket
Batch Operations	Run one action over millions of objects

Notifications reach Lambda, SQS, SNS, or EventBridge within seconds and are delivered at least once, so handlers must be idempotent. Replication requires versioning on both buckets and an IAM role.

Gotcha: Replication only copies objects written after the rule exists. Backfill existing objects with S3 Batch Replication.

Further reading

- [Amazon S3 User Guide](#)
- [AWS CLI s3 commands](#)
- [S3 bucket policy examples](#)

- S3 pricing