Cheatography

aws-s3-dva-co2 Cheat Sheet

by sevco-cymru via cheatography.com/173390/cs/45654/

Description

Infinitely scaling storage service.

Usage

Backup and Storage

Disaster Recovery

Archiving

Hosting

Data Lake and Analytics

Static Websites

Encryption (at rest)

Server Side Encryption (SSE-)	
SSE-S3	Only AWS has access to the keys
	AES-256 standard
	Header: x-amz-server-side-encryption:AES256
	Enabled by default
SSE-KMS	User managed keys
	Gives key control to user + Cloudtrail auditing
	Header: x-amz-server-side-encryption:aws-kms
	Can be limited by KMS Limits (quota can be increased)
	APIs - generatedatakey DecryptKMS
SSE-C	Custom key. Still Server Side
	AWS does not keep the key after creation
	Key is passed in header **https only**
Client Side Encryption	Uses the S3 client side encryption library
	Client fully manages the encryption cycle

Versioning

Versioning set at bucket level

If a key is overwritten a new key is created

If versioning is suspended previous versions are **not** deleted

Prior to versioning the v-id is `null`

Replication Steps

S3 management tab -> Replication Rules

Replication

Cross Region Replication	Compliance, lower latency access, x-acc replication
Same Region Replication	Prod ->Test replication, log aggregation
Replication can be set for all or some objects	
Versioning must be enabled for replication	
After versioning only new objects have v-ids	Use 1x batch replication to replicate existing objects first time
Deletes	objects with version id are **not** replicated avoids malicious deletes
	Delete Marker replication must be enabled
Replication cannot be chained	e.g. B1->B2->B3 needs to be set up as: B1->B2 & B1->B3 **cannot do** B1-> B2 / B3

Pre-Signed URLs

Generate using S3console	TTL 1m - 720m
Generate using AWS Cli	TTL default 3600s max 604800s

Pre-Signed URLs (cont)

Users with the url inherit the generating user's permissions

Use to give one off access to someone else e.g. temp access to a file

Naming

Name must be **globally** unique	All lower case
Between 3 and 63 characters	Must **not** be an ip address
Must start with a letter or number	**No** underscores (_)
Prefix restriction: **xn--**	Suffix restriction: -**s3alias**

Security

User based -	IAM policies
Resource Based -	S3 bucket level policies (most common) Object ACL - fine grained. Can disable Bucket ACL - can disable

If a bucket should never be public leave **All Public Access** as blocked
This can be set at the account level

Bucket Policy

Resource Block:	Bucket / object
Effect:	**Allow** / **Deny**
Action:	API actions affected by Effect
Principle:	Account or user to apply policy to

Use bucket policies to ...
Grant public access to a bucket
Force encryption @ upload
Grant access to another account

Lifecycle Rules w/ S3 Analytics

Transition action allows transition to different classes

Expiration action (deletes)

Can specify rules for a **prefix** or **tag**

S3 analytics allow to decide best strategy (works on Standard or Standard IA)

Analytics report is updated daily

S3 Event Notifications

Triggers:	ObjectCreated ObjectRemoved ObjectRestore ObjectReplication

Possible to filter on prefix or suffix

Available Events:	SNS SQS Lambda
Permissions -	SNS Resource Access policy SQS Resource Access policy Lambda Resource Access policy

S3 EventBridge

Add rules to the bridge

Allows access to 18+ services

Advanced filtering

Multiple Destinations for notification

Levarage EventBridge capabilities

Encryption (in flight)

SSL / TLS	Use https endpoint to force encryption

Force In-Transit Encyption using bucket policy

Add policy to refuse API calls without encryption headers	`effect: DENY`
	`condition:`
	`bool:`
	`"aws:SecureTransport":"false"`

n.b. Bucket policies are evaluated before defaults

MFA Deletes

Adds security around:	Permanently delete an object
	Suspend versioning

Only bucket owner (root) can enable

Enabled using CLI

```
> aws s3api put-bu cke t-v ers ioning < ...>
```

Delete is via cli

Keys (Identifying objects)

An S3 object is identified by its key:

s3://*<unique bucket name>/[<prefixes>]/object-name*

Everything in S3 is a key/object.

There is no concept of directories - these are prefixes

Storage Classes

Standard (STANDARD)	Default storage class. Can use with Intelligent Tiering to move to STANDARD_IA using S3 analytics
S3 Standard-IA (STANDARD_IA)	Long-lived, infrequently accessed data (once a month) with millisecond access
S3 One Zone-IA (ONEZONE_IA)	Lost if AZ is destroyed Recreatable, infrequently accessed data (once a month) with millisecond access
S3 Express One Zone (EXPRESS_ONEZONE)	Single-digit millisecond data access for latency-sensitive applications within a single AWS Availability Zone
S3 Glacier Instant Retrieval (GLACIER_IR)	Long-lived, archive data accessed once a quarter with millisecond access

Storage Classes (cont)

S3 Glacier Flexible Retrieval (GLACIER)	Long-lived archive data accessed once a year with retrieval times of minutes to hours
S3 Glacier Deep Archive (DEEP_ARCHIVE)	Long-lived archive data accessed less than once a year with retrieval times of hours Standard (12h) Bulk (48h)
Intelligent Tiering	Data with unknown, changing, or unpredictable access patterns
Reduced Redundancy Storage (REDUCED_REDUNDANCY) Not recommended	Noncritical, frequently accessed data with millisecond access

Lifecycle Rules used to move objects between classes

Storage Class Config Options

STANDARD	-	-
S3 Standard-IA (STANDARD_IA)	-	Per object >=128kb (monitoring + auto-tiering))
ONEZONE_IA	30 days+ (min. storage)	Per-GB fees (retrieval)
GLACIER_IR	90 days+	Per-GB fees (retrieval)
GLACIER (Flexible)	90 days+	Per-GB fees (retrieval)
DEEP_ARCHIVE	180 days+	Per-GB fees (retrieval)

Minimum billable object size for Standard IA / One Zone IA / Glacier IR

S3 Access Logs

Log actions into a **different** bucket

Never set monitored-bucket=log-bucket (creates loop + $$$)

s3 Cross Origin Resource Sharing

If x-origin requests are required need correct CORS headers

Can allow for specific files or *

Setting written in json

S3 Access Points

Placeholder