


## Lifecycle management in S3:



## VERSIONING

# Combining Lifecycle Management with Versioning

You can use lifecycle management to **move different versions** of objects to **different storage tiers**.

# 3 Tips for Lifecycle Management

- ✓ Automates moving objects between different storage tiers.
- ✓ Can be used in conjunction with versioning.
- ✓ Can be applied to current versions and previous versions.

## S3 Object Lock and Glacier Vault Lock:

## S3 Object Lock

You can use S3 Object Lock to store objects using a **write once, read many (WORM)** model. It can help prevent objects from being deleted or modified for a fixed amount of time or indefinitely.

You can use **S3 Object Lock** to meet regulatory requirements that require WORM storage, or add an extra layer of protection against object changes and deletion.

### S3 OBJECT LOCK MODES

## Governance Mode

In governance mode, **users can't overwrite or delete an object version or alter its lock settings** unless they have special permissions.

With governance mode, you protect objects against being deleted by most users, but you can still grant some users **permission to alter the retention settings** or delete the object if necessary.

### S3 OBJECT LOCK MODES

## Compliance Mode

In compliance mode, **a protected object version can't be overwritten or deleted by any user**, including the root user in your AWS account. When an object is locked in compliance mode, its retention mode can't be changed and its retention period can't be shortened. Compliance mode ensures an object version **can't be overwritten or deleted** for the duration of the retention period.

## Retention Periods

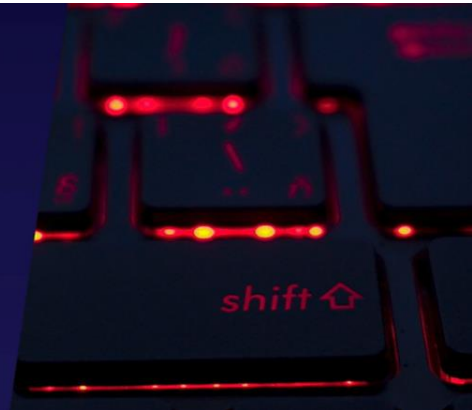
A retention period **protects an object version for a fixed amount of time**. When you place a retention period on an object version, Amazon S3 stores a timestamp in the object version's metadata to indicate when the retention period expires.

After the retention period expires, the object version can be **overwritten or deleted** unless you also placed a legal hold on the object version.



## Legal Holds

S3 Object Lock also enables you to place a legal hold on an object version. Like a retention period, a legal hold **prevents an object version from being overwritten or deleted**. However, a legal hold doesn't have an associated retention period and remains in effect until removed. Legal holds can be freely placed and removed by any user who has the `s3:PutObjectLegalHold` permission.



## Glacier Vault Lock

S3 Glacier Vault Lock allows you to **easily deploy and enforce compliance controls for individual S3 Glacier vaults with a vault lock policy**. You can specify controls, such as WORM, in a vault lock policy and lock the policy from future edits. Once locked, the policy can no longer be changed.



## 3 Object Lock Tips

- ✓ Use **S3 Object Lock** to store objects using a write once, read many (WORM) model.
- ✓ Object Lock can be on **individual objects** or applied **across the bucket** as a whole.
- ✓ Object Lock comes in two modes: **governance mode** and **compliance mode**.



# S3 Object Lock Modes

## 1 Compliance Mode

With **compliance mode**, a protected object version can't be overwritten or deleted by any user, including the root user in your AWS account.

## 2 Governance Mode

With **governance mode**, users can't overwrite or delete an object version or alter its lock settings unless they have special permissions.

## S3 Glacier Vault Lock Exam Tips

- ✓ S3 Glacier Vault Lock allows you to **easily deploy** and **enforce compliance controls** for individual S3 Glacier vaults with a vault lock policy.
- ✓ You can **specify controls, such as WORM, in a vault lock policy and lock the policy from future edits**. Once locked, the policy can no longer be changed.

## Types of Encryption

### 1 Encryption in Transit

- SSL/TLS
- HTTPS

### 2 Encryption at Rest: Server-Side Encryption

- **SSE-S3**: S3-managed keys, using AES 256-bit encryption
- **SSE-KMS**: AWS Key Management Service-managed keys
- **SSE-C**: Customer-provided keys

### 3 Encryption at Rest: Client-Side Encryption

You encrypt the files yourself before you upload them to S3.



# Enforcing Server-Side Encryption

## Two Ways to Do It



### Console

Select the encryption setting on your S3 bucket. The easiest way is just a checkbox in the console.



### Bucket Policy

You can also enforce encryption using a bucket policy. This method sometimes comes up in the exam.

# Enforcing Server-Side Encryption

1

#### **x-amz-server-side-encryption**

If the file is to be encrypted at upload time, the **x-amz-server-side-encryption** parameter will be included in the request header.

2

#### **Two Options**

**x-amz-server-side-encryption: AES256**  
(SSE-S3 — S3-managed keys)

**x-amz-server-side-encryption: aws:kms**  
(SSE-KMS — KMS-managed keys)

3

#### **PUT Request Header**

When this parameter is included in the header of the PUT request, it tells S3 to encrypt the object at the time of upload, using the specified encryption method.

# Exam Tips



## Encryption in Transit

- SSL/TLS
- HTTPS



## Client-Side Encryption

You encrypt the files yourself before you upload them to S3.



## Encryption at Rest: SSE

- Server-side encryption
- SSE-S3 (AES 256-bit)
- SSE-KMS
- SSE-C



## Enforcing Encryption with a Bucket Policy

A bucket policy can deny all PUT requests that don't include the

encryption



Next lesson

Optimizing S3 Performance