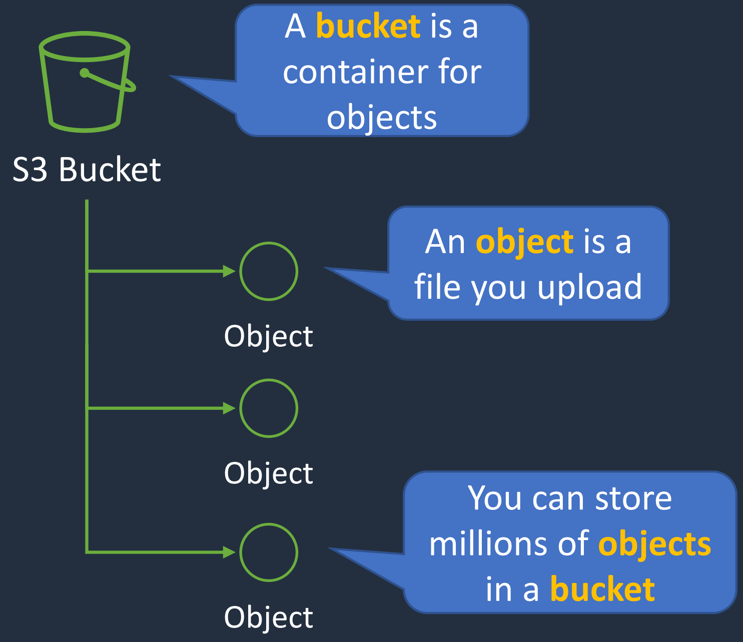
Amazon S3 Overview

Amazon Simple Storage Service (S3)



A **bucket** is a container for objects

An **object** is a file you upload

You can store millions of **objects** in a **bucket**

Accessing objects in a bucket:

```
https://bucket.s3.aws-region.amazonaws.com/key
https://s3.aws-region.amazonaws.com/bucket/key
```

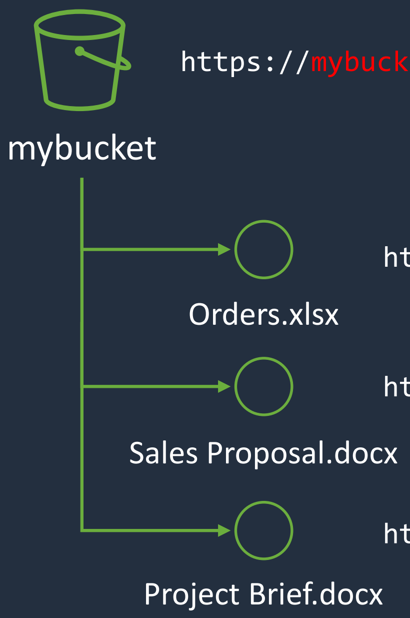
The **HTTP protocol** is used with a **REST API** (e.g. GET, PUT, POST, SELECT, DELETE)

DigitalCloud TRAINING

Amazon Simple Storage Service (S3)

- You can store any type of file in S3
- Files can be anywhere from 0 bytes to 5 TB
- There is unlimited storage available
- S3 is a universal namespace so **bucket names** must be **unique globally**
- However, you create your buckets within a **REGION**
- It is a best practice to create buckets in regions that are physically closest to your users to reduce latency
- There is no hierarchy for objects within a bucket
- Delivers strong read-after-write consistency

Buckets, Folders, and Objects



https://mybucket.s3-website-us-west-1.amazonaws.com

https://mybucket.s3.us-west-1.amazonaws.com/Orders.xlsx

https://mybucket.s3.us-west-1.amazonaws.com/Project+Alpha/Sales+Proposal.docx

https://mybucket.s3.us-west-1.amazonaws.com/Project+Alpha/Project+Brief.docx

The object name is the **Key** the data is the **Value**

A **Folder** is a shared **prefix** for grouping objects

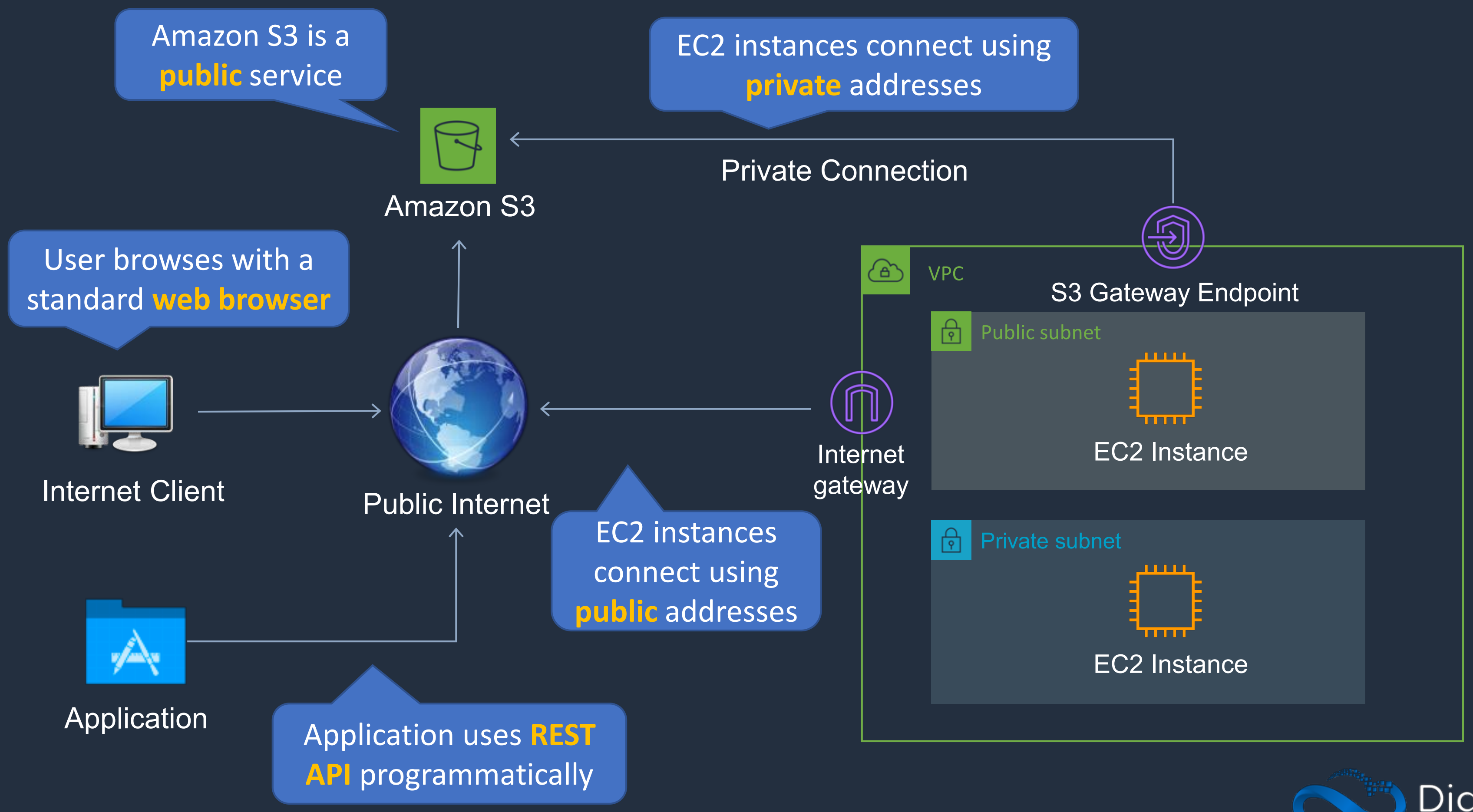
DigitalCloud TRAINING

Buckets, Folders, and Objects

- Folders **can** be created within folders
- Buckets **cannot** be created within other buckets
- An objects consists of:
  - Key (the name of the object)
  - Version ID
  - Value (actual data)
  - Metadata
  - Subresources
  - Access control information

DigitalCloud TRAINING

Accessing Amazon S3



Amazon S3 is a **public** service

EC2 instances connect using **private** addresses

User browses with a standard **web browser**

EC2 instances connect using **public** addresses

Application uses **REST API** programmatically

Amazon S3 Storage Classes

Durability and Availability in S3

Durability

Durability is protection against:

- Data loss
- Data corruption
- S3 offers 11 9s durability (99.999999999)

If you store 10 million objects, then you expect to lose one object every 10,000 years!

Availability

Availability is a measurement of:

- The amount of time the data is available to you
- Expressed as a percent of time per year
- E.g. 99.99%

DigitalCloud TRAINING

S3 Storage Classes

	S3 Standard	S3 Intelligent Tiering	S3 Standard-IA	S3 One Zone-IA	S3 Glacier Instant Retrieval	S3 Glacier Flexible Retrieval	S3 Glacier Deep Archive
Designed for durability	99.999999999%	99.999999999%	99.999999999%	99.999999999%	99.999999999%	99.999999999%	99.999999999%
Designed for availability	99.99%	99.9%	99.9%	99.5%	99.9%	99.99%	99.99%
Availability SLA	99.9%	99%	99%	99%	99%	99.9%	99.9%
Availability Zones	≥3	≥3	≥3	1	≥3	≥3	≥3
Minimum capacity charge per object	N/A	N/A	128KB	128KB	128KB	40KB	40KB
Minimum storage duration charge	N/A	N/A	30 days	30 days	90 days	90 days	180 days
Retrieval fee	N/A	N/A	Per GB retrieved	Per GB retrieved	Per GB retrieved	Per GB retrieved	Per GB retrieved
First byte latency	milliseconds	milliseconds	milliseconds	milliseconds	milliseconds	minutes or hours	hours
Storage type	Object	Object	Object	Object	Object	Object	Object
Lifecycle transitions	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Amazon S3 Lifecycle Policies

S3 Lifecycle Management

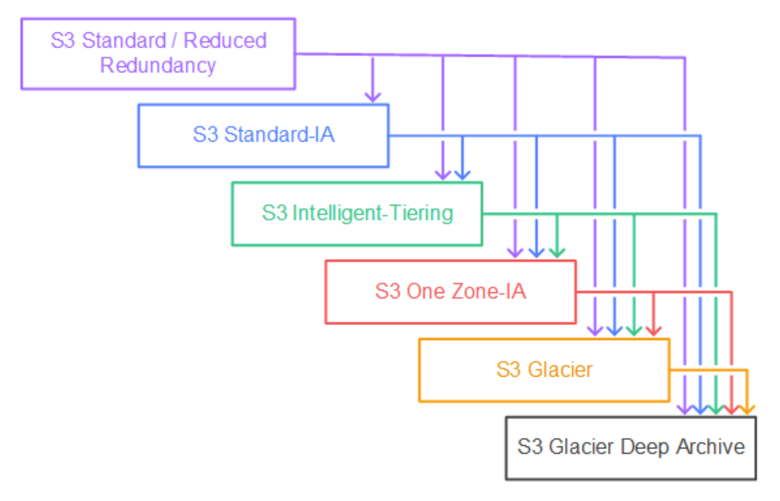
There are two types of actions:

- **Transition actions** - Define when objects transition to another storage class
- **Expiration actions** - Define when objects expire (deleted by S3)

S3 LM: Supported Transitions

You can transition from the following:

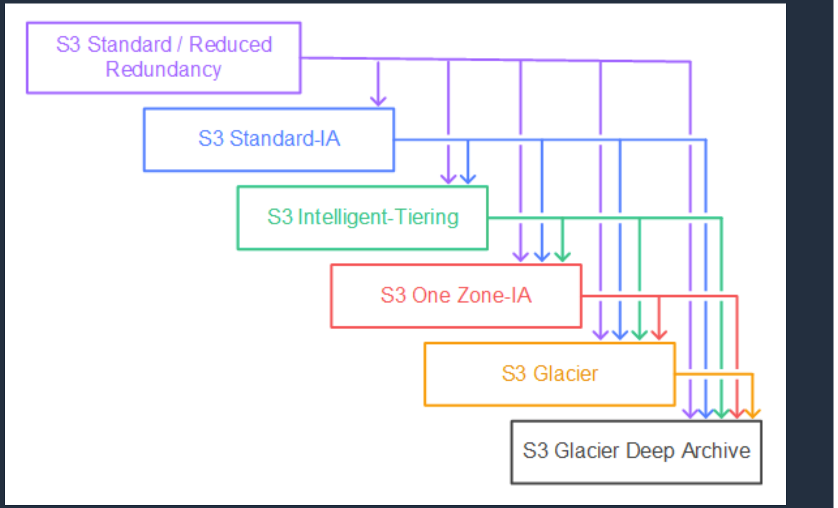
- The S3 Standard storage class to any other storage class
- Any storage class to the S3 Glacier or S3 Glacier Deep Archive storage classes
- The S3 Standard-IA storage class to the S3 Intelligent-Tiering or S3 One Zone-IA storage classes
- The S3 Intelligent-Tiering storage class to the S3 One Zone-IA storage class
- The S3 Glacier storage class to the S3 Glacier Deep Archive storage class



S3 LM: Unsupported Transitions

You can't transition from the following:

- Any storage class to the S3 Standard storage class
- Any storage class to the Reduced Redundancy storage class
- The S3 Intelligent-Tiering storage class to the S3 Standard-IA storage class
- The S3 One Zone-IA storage class to the S3 Standard-IA or S3 Intelligent-Tiering storage classes

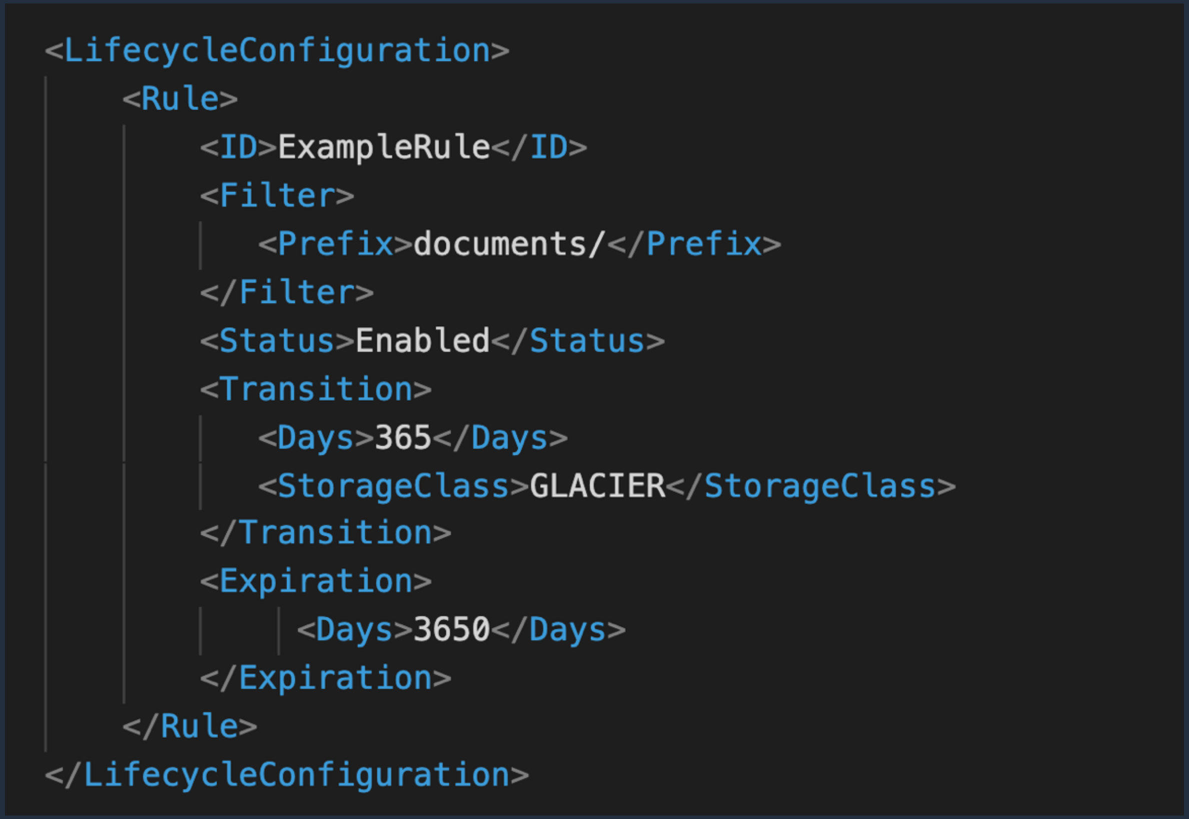


S3 Lifecycle Management

- Can create a lifecycle policy through the console or CLI/API
- When configured using the CLI/API an XML or JSON file must be supplied
- API actions to create/update/delete lifecycle policies:
  - PutBucketLifecycleConfiguration - Creates a new lifecycle configuration for the bucket or replaces an existing lifecycle configuration
  - GetBucketLifecycleConfiguration - Returns the lifecycle configuration information set on the bucket
  - DeleteBucketLifecycle - Deletes the lifecycle configuration from the specified bucket

DigitalCloud TRAINING

Example S3 Lifecycle Policy (XML)



```xml
<LifecycleConfiguration>
    <Rule>
        <ID>ExampleRule</ID>
        <Filter>
           <Prefix>documents/</Prefix>
        </Filter>
        <Status>Enabled</Status>
        <Transition>
           <Days>365</Days>
           <StorageClass>GLACIER</StorageClass>
        </Transition>
        <Expiration>
               <Days>3650</Days>
        </Expiration>
    </Rule>
</LifecycleConfiguration>
```

Configure Replication and Lifecycle

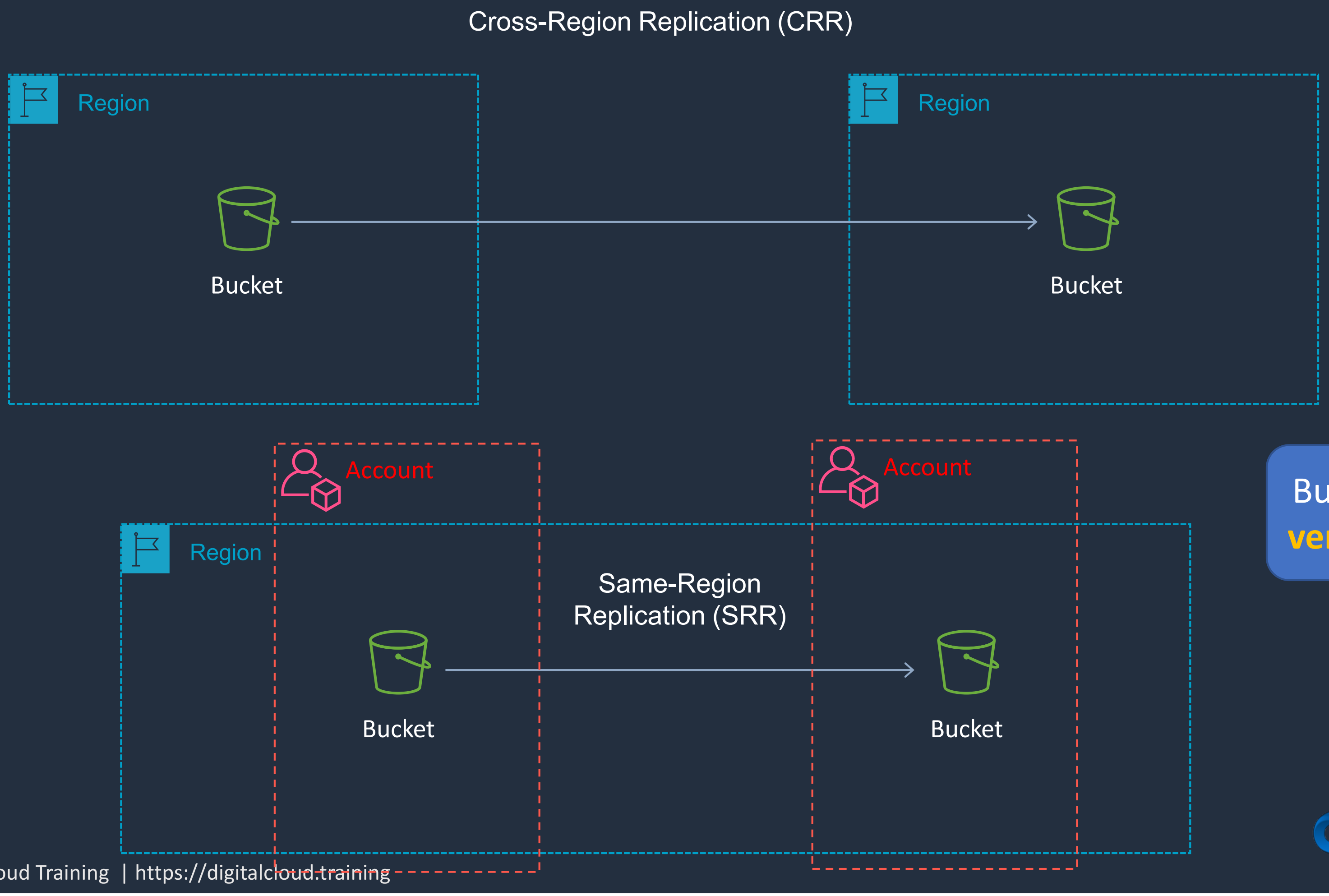
S3 Versioning and Replication

S3 Versioning

- Versioning is a means of keeping multiple variants of an object in the same bucket
- Use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket
- Versioning-enabled buckets enable you to recover objects from accidental deletion or overwrite

DigitalCloud TRAINING

S3 Replication



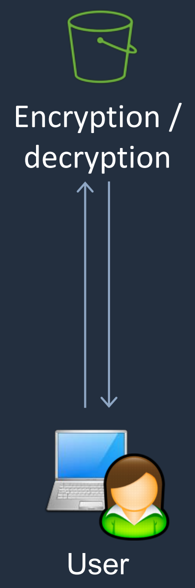
Buckets must have **versioning** enabled

Amazon S3 Encryption

Amazon S3 Encryption

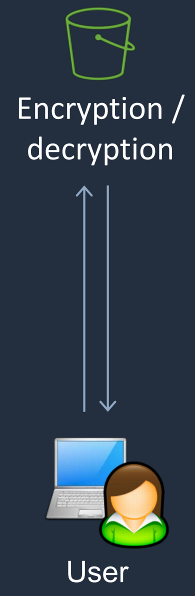
Server-side encryption with S3 managed keys (SSE-S3)

- S3 managed keys
- Unique object keys
- Master key
- AES 256



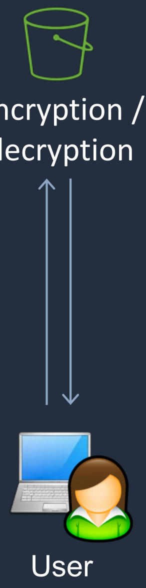
Server-side encryption with AWS KMS managed keys (SSE-KMS)

- KMS managed keys
- Can be AWS managed keys
- Or customer managed KMS keys



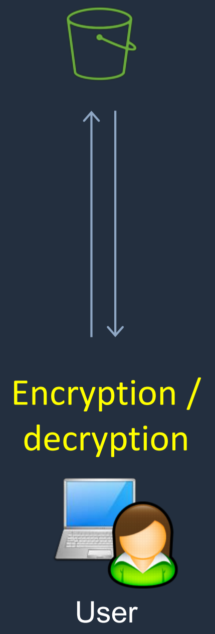
Amazon S3 Encryption

Server-side encryption with client provided keys (SSE-C)



- Client managed keys
- Not stored on AWS

Client-side encryption

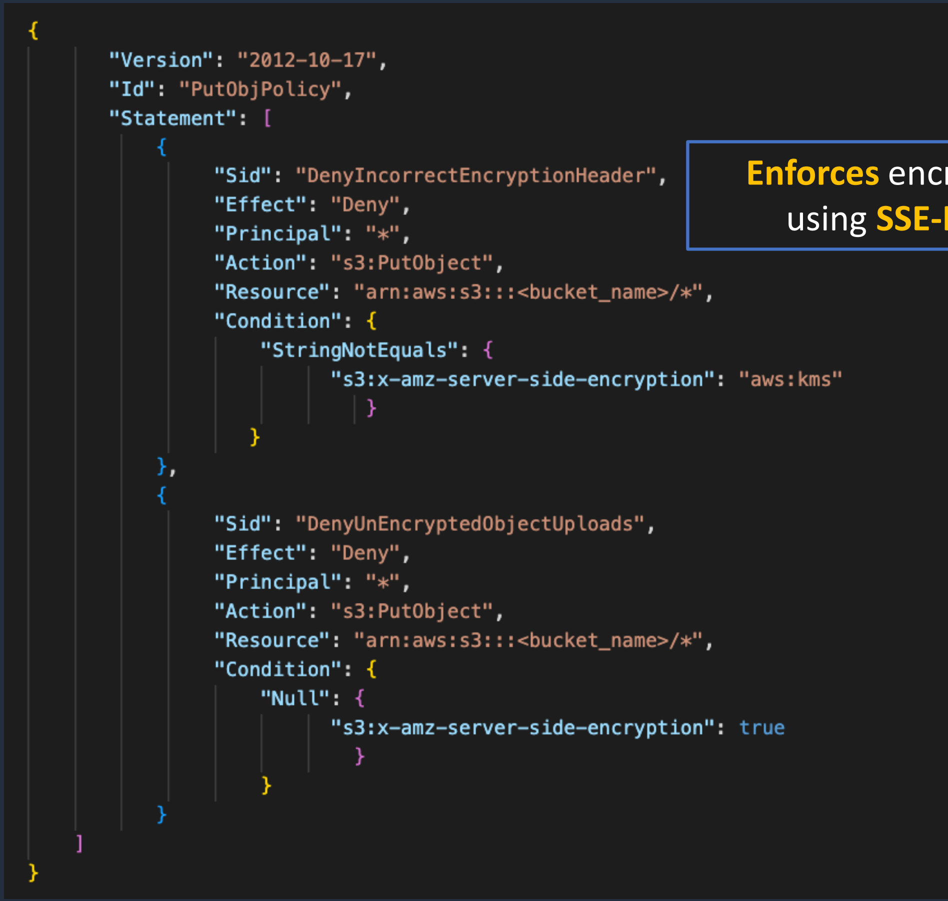


- Client managed keys
- Not stored on AWS
- Or you can use a KMS Key

Amazon S3 Default Encryption

- All Amazon S3 buckets have encryption configured by default
- All new object uploads to Amazon S3 are automatically encrypted
- There is no additional cost and no impact on performance
- Objects are automatically encrypted by using server-side encryption with Amazon S3 managed keys (SSE-S3)
- To encrypt existing unencrypted Amazon S3 objects, you can use Amazon S3 Batch Operations
- You can also encrypt existing objects by using the CopyObject API operation or the copy-object AWS CLI command

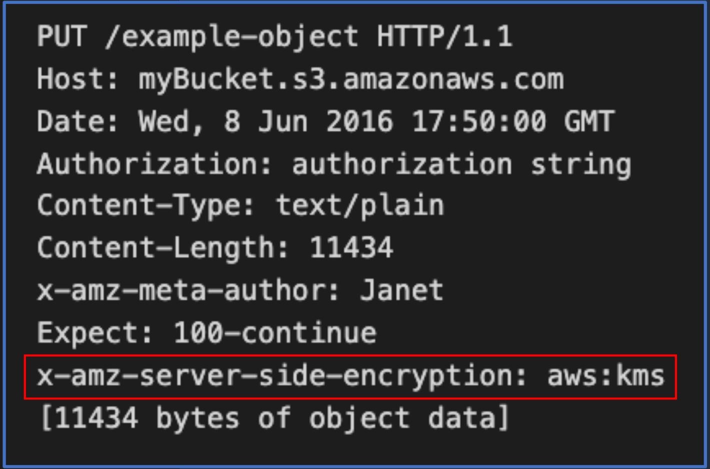
Enforce Encryption with Bucket Policy



```
{
    "Version": "2012-10-17",
    "Id": "PutObjPolicy",
    "Statement": [
        {
            "Sid": "DenyIncorrectEncryptionHeader",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::<bucket_name>/*",
            "Condition": {
                "StringNotEquals": {
                    "s3:x-amz-server-side-encryption": "aws:kms"
                }
            }
        },
        {
            "Sid": "DenyUnEncryptedObjectUploads",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::<bucket_name>/*",
            "Condition": {
                "Null": {
                    "s3:x-amz-server-side-encryption": true
                }
            }
        }
    ]
}
```

Enforces encryption using **SSE-KMS**

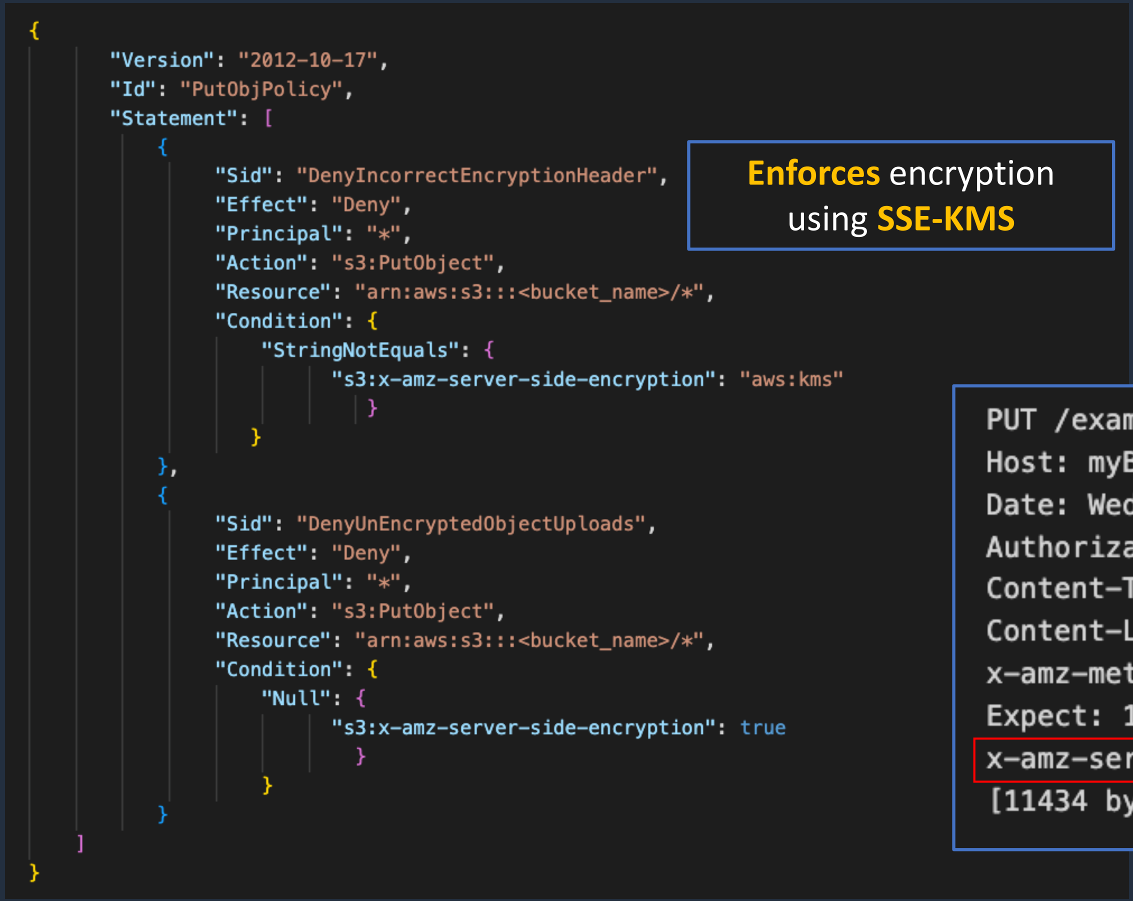
Example PUT request



```
PUT /example-object HTTP/1.1
Host: myBucket.s3.amazonaws.com
Date: Wed, 8 Jun 2016 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: 11434
x-amz-meta-author: Janet
Expect: 100-continue
x-amz-server-side-encryption: aws:kms
[11434 bytes of object data]
```

Enforce Encryption with AWS KMS

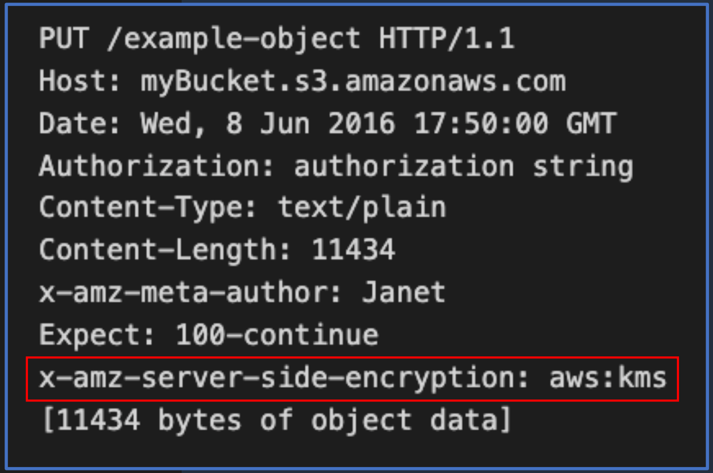
Enforce Encryption with Bucket Policy



```
{
    "Version": "2012-10-17",
    "Id": "PutObjPolicy",
    "Statement": [
        {
            "Sid": "DenyIncorrectEncryptionHeader",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::<bucket_name>/*",
            "Condition": {
                "StringNotEquals": {
                    "s3:x-amz-server-side-encryption": "aws:kms"
                }
            }
        },
        {
            "Sid": "DenyUnEncryptedObjectUploads",
            "Effect": "Deny",
            "Principal": "*",
            "Action": "s3:PutObject",
            "Resource": "arn:aws:s3:::<bucket_name>/*",
            "Condition": {
                "Null": {
                    "s3:x-amz-server-side-encryption": true
                }
            }
        }
    ]
}
```

Enforces encryption using **SSE-KMS**

Example PUT request



```
PUT /example-object HTTP/1.1
Host: myBucket.s3.amazonaws.com
Date: Wed, 8 Jun 2016 17:50:00 GMT
Authorization: authorization string
Content-Type: text/plain
Content-Length: 11434
x-amz-meta-author: Janet
Expect: 100-continue
x-amz-server-side-encryption: aws:kms
[11434 bytes of object data]
```

S3 Presigned URLs

S3 Presigned URLs

AWS S3 CLI command to generate a presigned URL

```
$ aws s3 presign s3://dct-data-bucket/cool_image.jpeg
```

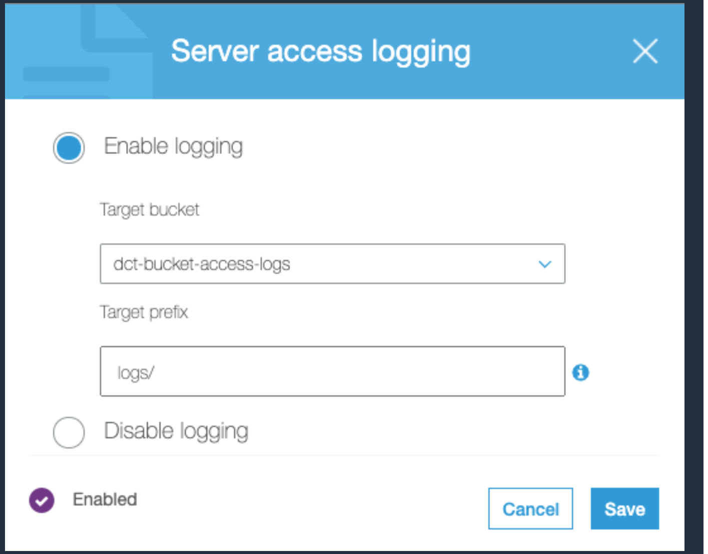
```
$ https://dct-data-bucket.s3.ap-southeast-2.amazonaws.com/cool_image.jpeg?X-Amz-
Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIA3KSVPHP6MAHNW5YH%2F20200909%2Fap-southeast-
2%2Fs3%2Faws4_request&X-Amz-Date=20200909T053538Z&X-Amz-Expires=3600&X-Amz-
SignedHeaders=host&X-Amz-
Signature=8b74653beee371da07a73dfdb4ff6883742383afa528aecd5c95c326c97764db
```

This is the response; the URL expires after 1 hour

Server Access Logging

Server Access Logging

- Provides detailed records for the requests that are made to a bucket
- Details include the requester, bucket name, request time, request action, response status, and error code (if applicable)
- Disabled by default
- Only pay for the storage space used
- Must configure a separate bucket as the destination (can specify a prefix)
- Must grant write permissions to the Amazon S3 Log Delivery group on destination bucket



Server access logging

● Enable logging

Target bucket: dct-bucket-access-logs

Target prefix: logs/

○ Disable logging

✓ Enabled