COMP5349– Cloud Computing Week 2: Cloud Storage

Dr. Ying Zhou

The University of Sydney

Table of Contents

COMMONWEALTH OF AUSTRALIA

Copyright Regulations 1969

WARNING

This material has been reproduced and communicated to you by or on behalf of the **University of Sydney** pursuant to Part VB of the Copyright Act 1968 (the Act).

The material in this communication may be subject to copyright under the Act. Any further reproduction or communication of this material by you may be the subject of copyright protection under the Act.

Do not remove this notice

01 Cloud Storage Types

02 Amazon S3: A distributed object store

03 S3 Common Use Cases

04 S3 Storage Options

05 S3 Versioning

06 S3 Replication

Cloud Storage Types

Storage as SaaS

- Many cloud storage services act like an extension to the local hard disk
- They provide similar features as local file system
- Plus some "cloud" features
  - Sharing
  - Collaborative editing
  - Versioning

- Google Drive
- OneDrive
- Dropbox
- iCloud

Storage as IaaS/PaaS

- There are also storage services provided as independent "infrastructure" or as part of the other infrastructure or platform
  - AWS S3
  - Other AWS storage types that can be mounted to EC2 instances
  - Azure storage
  - Google Cloud Storage
- They have much less end user UI features but allows more control on other aspects
- They are usually more cost-effective

AWS Data Storage Services

Service	Access	Maximum storage volume	Latency	Storage Cost
S3	AWS API (SDKs, CLI), third-party tools	unlimited	High	Very low
EBS (SSD)	Attached to an EC2 instance via network	16 TiB	Low	Low
EC2 Instance Store (SSD)	Attached to an EC2 instance directly	305 TB	Very low	Very low
EFS	NFSv4.1, for example from an EC2 instance or on-premises	Unlimited	Medium	Medium

AWS S3: A distributed object store

File store and Object store

- In a typical file storage system, data are managed in a hierarchy consisting of folders and files
  - The file is the representation of the data
- In an ***object storage***, data is stored as objects
- Each object consists of
  - A *globally unique identifier*
  - Some *metadata*
    - Access control, object size, etc
  - The content itself
    - Which is similar to a file

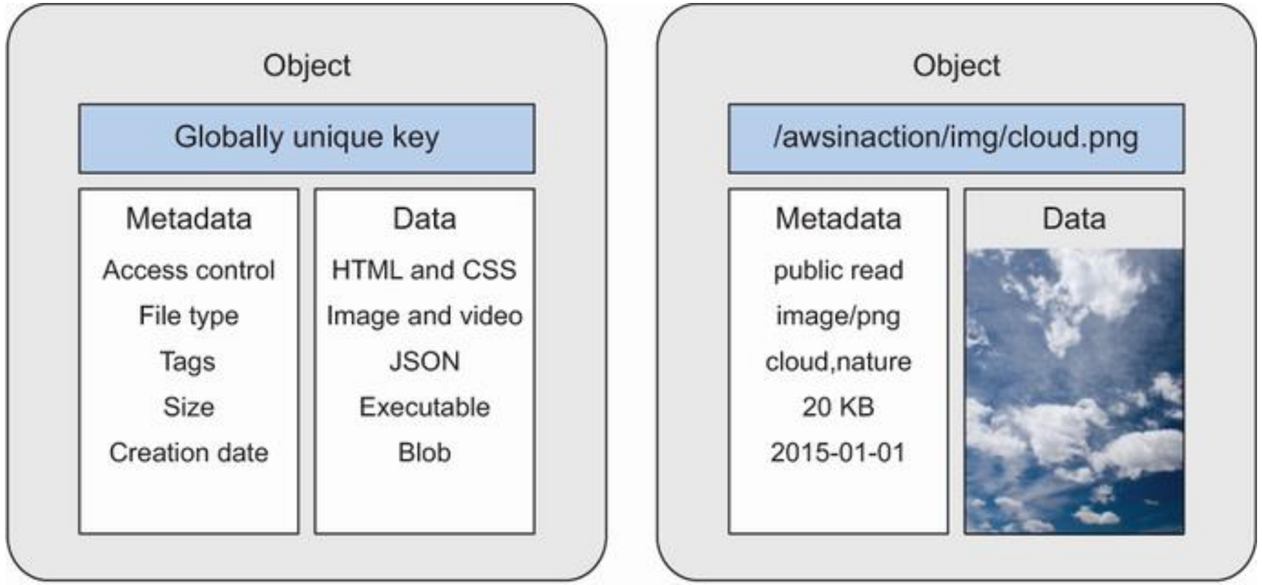


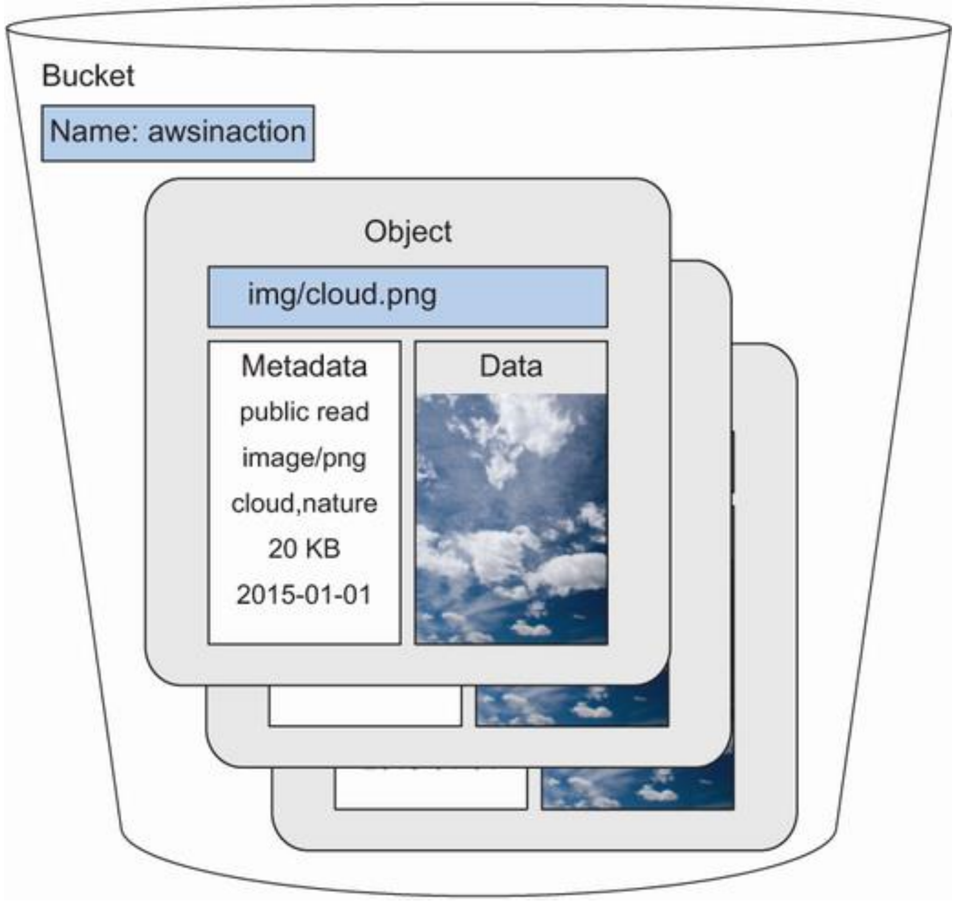
Figure 7.1 Objects stored in an object store have three parts: a unique ID, metadata describing the content, and the content itself (such as an image).

S3 – Simple Storage Service

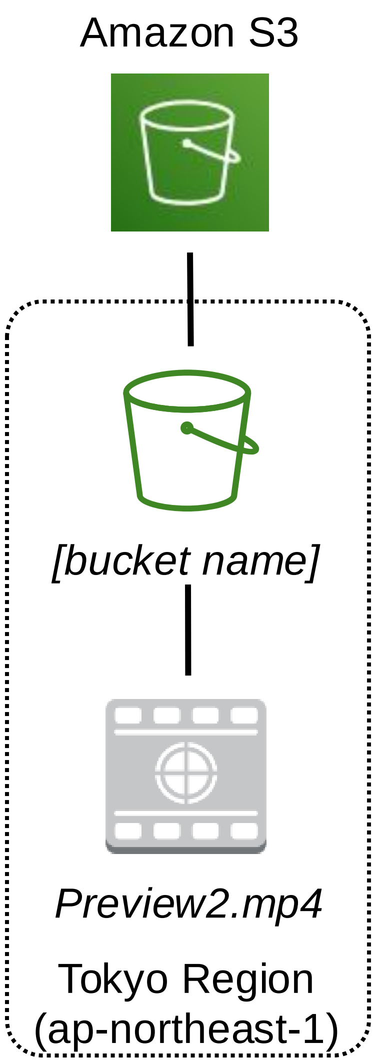
- One of the oldest services provided by AWS
  - Part of the first three services released by AWS in 2006
  - It is classified as a distributed object store
- Each data is an object with a unique identifier (key)
- Data can be uploaded and retrieved through HTTPS as objects via APIs
- Local file vs. S3 object
  - Every "file" we store in S3 becomes an object.
  - The content part is the same
  - The difference lies in how data is organized, accessed and the metadata

Data organization: S3 buckets

- Any user will have a lot of objects to store
- They can be organized into buckets
  - a bucket is a container for objects.
- S3 uses a flat structure consists of buckets and objects
  - There is no further organization unit within a bucket
  - Different to hierarchical file structure
- An account can have many buckets
  - Each bucket needs to have a globally *unique* name
  - This helps to make the object identifier unique



Amazon S3 bucket URLs (two styles)



To upload your data:

1. Create a **bucket** in an AWS Region.
2. Upload almost any number of **objects** to the bucket.

Bucket path-style URL endpoint:

https://s3.ap-northeast-1.amazonaws.com/bucket-name

Region code **Bucket name**

Bucket virtual hosted-style URL endpoint:

https:// bucket-name.s3-ap-northeast-1.amazonaws.com

Bucket name **Region code**

The default region is “us-east-1”
which can be omitted

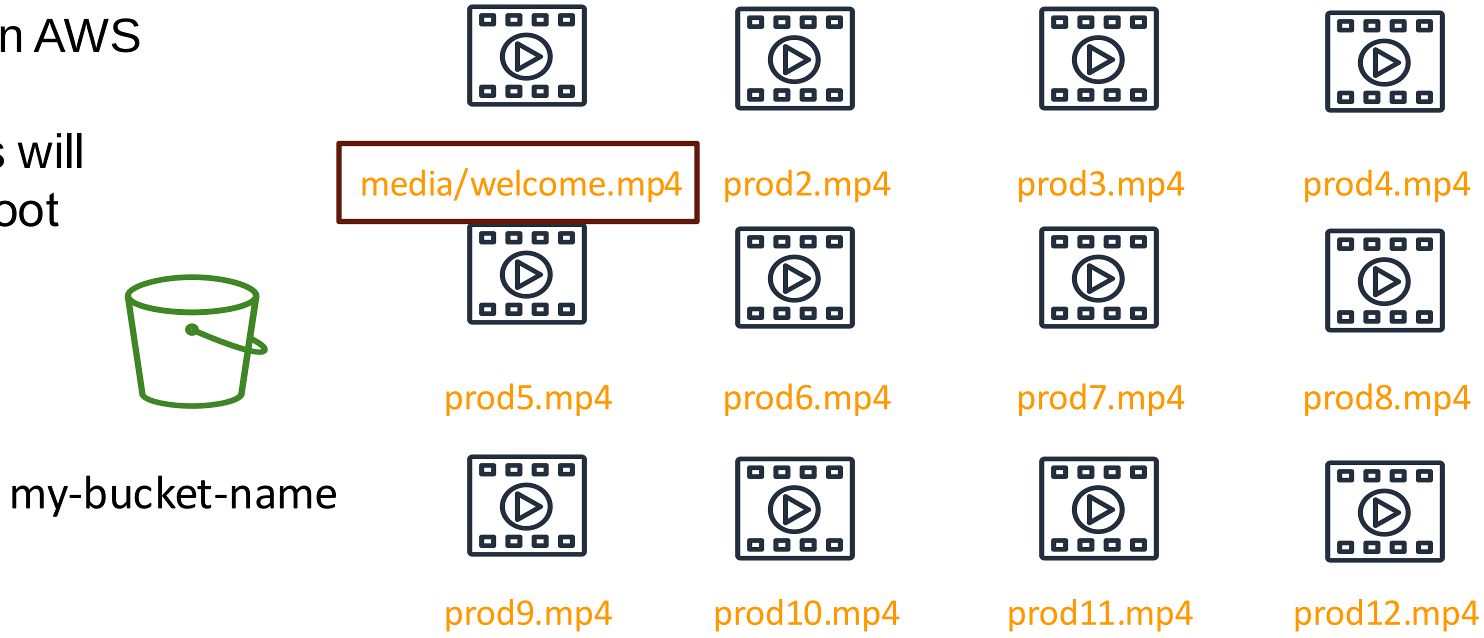
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

AWS S3 object key name and URL

- S3 provides pseudo directory structure to mimic the file system organization
  - The directory name is the prefix of the object key name

https:// my-bucket-nam.s3-ap-northeast-1.amazonaws.com/media/welcome.mp4

“media” will be treated like a directory in AWS S3 console
All other objects will appear at the “root directory” level



Pseudo folder structure in an S3 bucket

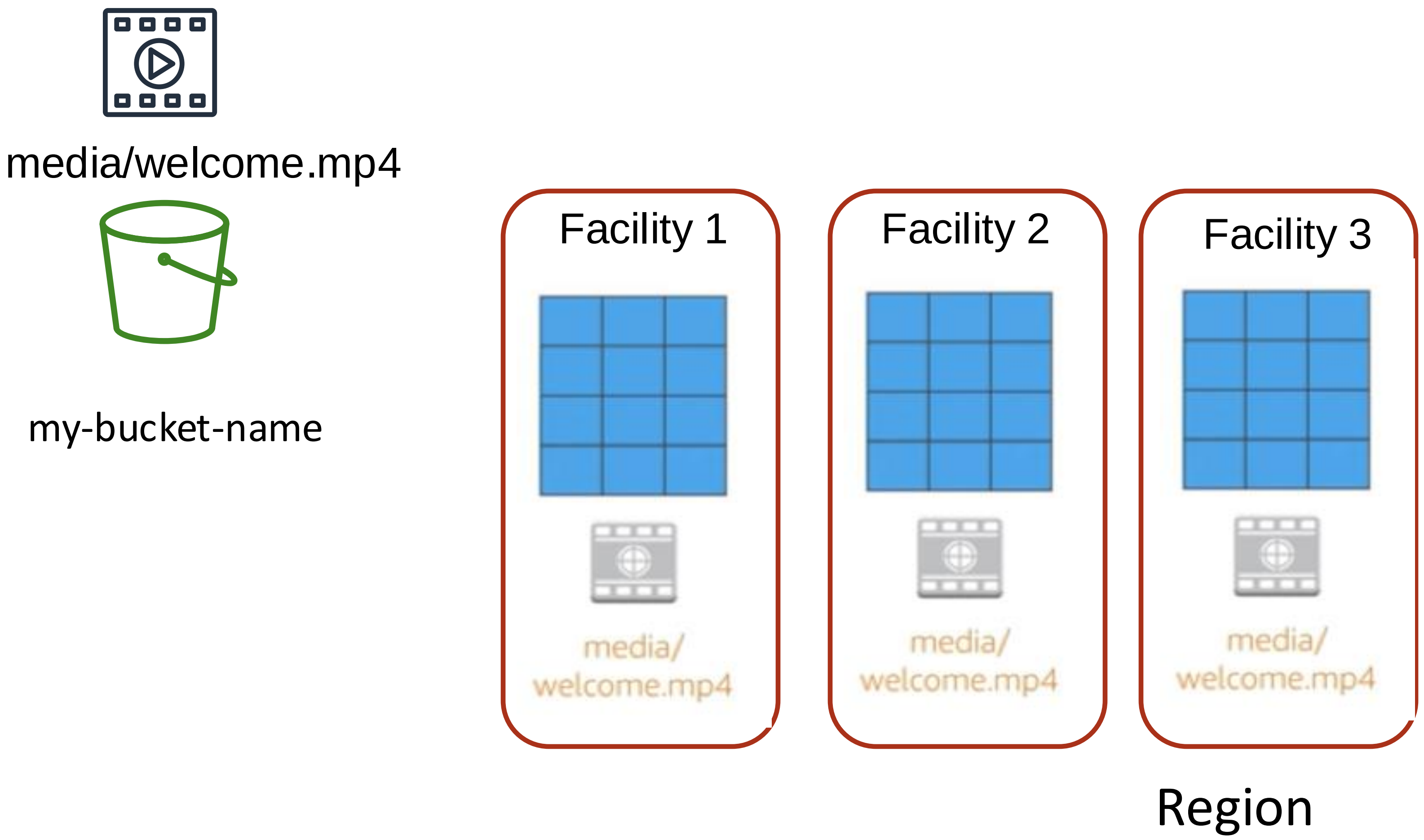
The following objects are in a bucket named **graphics-bucket**.

```
photos/2022/catpiano.jpg
photos/2022/catonphone.jpg
photos/2022/ninepuppies.png
photos/2021/lakefront.png
photos/2021/coveredbridge.png
photos/2021/openairmarket.jpg
video-source/9984.mp4
video-source/9918.mp4
video-source/18446.mp4
```

A **GET** query with the prefix **photos/2022** returns the following objects:

graphics-bucket/**photos/2022/catpiano.jpg**
graphics-bucket/**photos/2022/catonphone.jpg**
graphics-bucket/**photos/2022/ninepuppies.png**

Data is redundantly stored in the Region



media/welcome.mp4

my-bucket-name

Facility 1

media/
welcome.mp4

Facility 2

media/
welcome.mp4

Facility 3

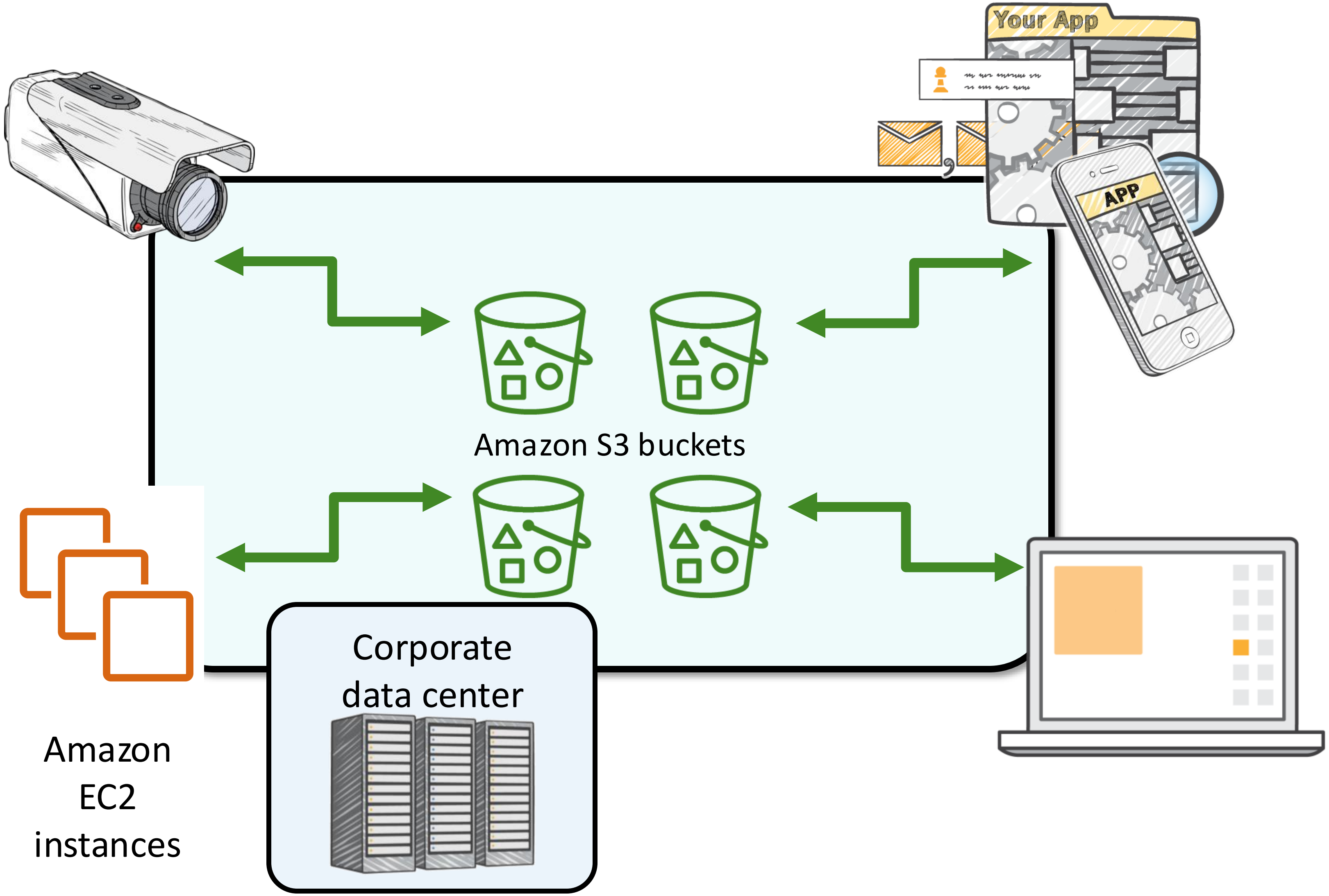
media/
welcome.mp4

Region

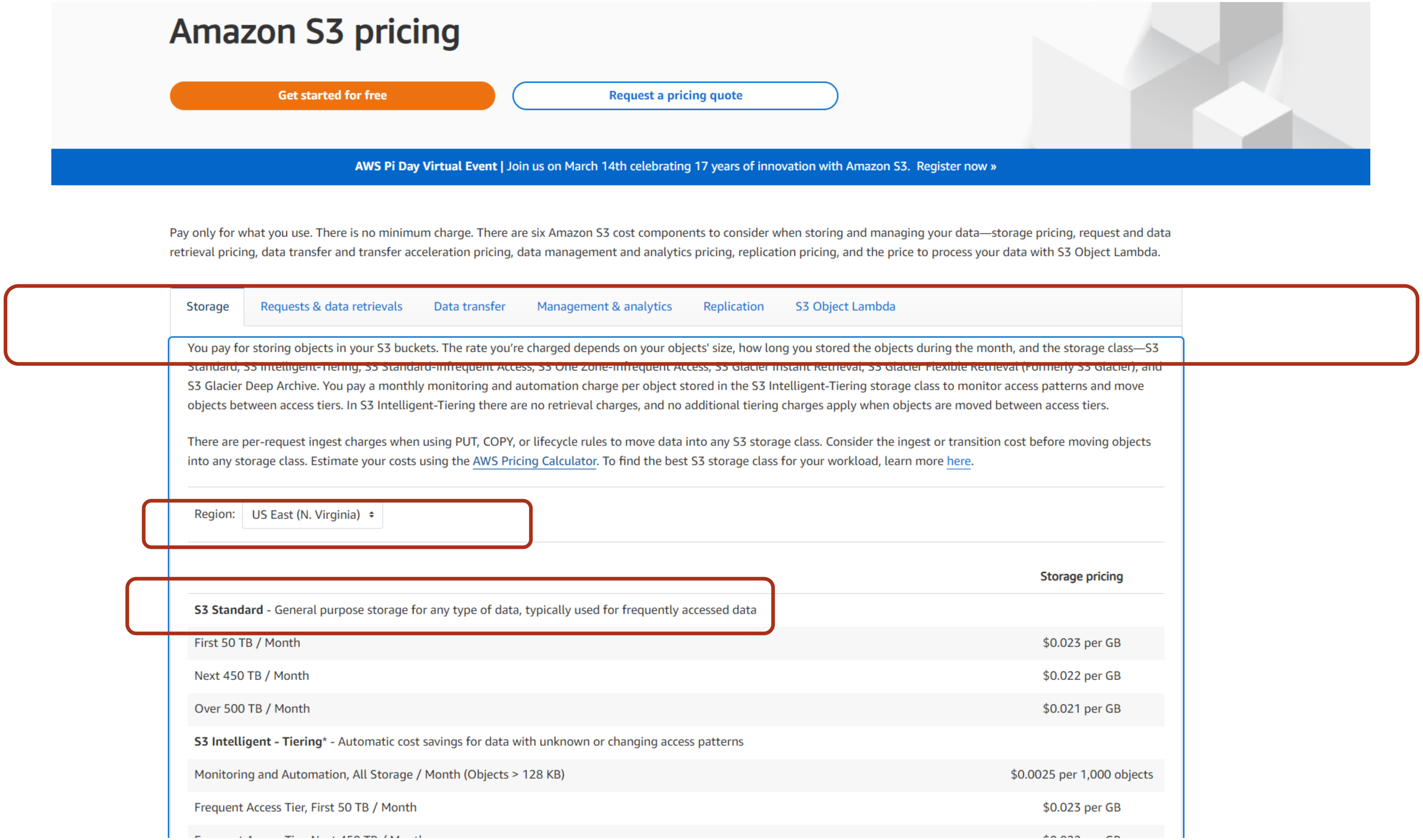
© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon S3 common scenarios

- Backup and storage
- Application hosting
- Media hosting
- Software delivery



© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.



Amazon S3 pricing

AWS Pi Day Virtual Event | Join us on March 14th celebrating 17 years of innovation with Amazon S3. Register now »

Pay only for what you use. There is no minimum charge. There are six Amazon S3 cost components to consider when storing and managing your data—storage pricing, request and data retrieval pricing, data transfer and transfer acceleration pricing, data management and analytics pricing, replication pricing, and the price to process your data with S3 Object Lambda.

You pay for storing objects in your S3 buckets. The rate you're charged depends on your objects' size, how long you stored the objects during the month, and the storage class—S3 Standard, S3 Intelligent-Tiering, S3 Standard-Infrequent Access, S3 One Zone-Infrequent Access, S3 Glacier Instant Retrieval, S3 Glacier Flexible Retrieval (Formerly S3 Glacier), and S3 Glacier Deep Archive. You pay a monthly monitoring and automation charge per object stored in the S3 Intelligent-Tiering storage class to monitor access patterns and move objects between access tiers. In S3 Intelligent-Tiering there are no retrieval charges, and no additional tiering charges apply when objects are moved between access tiers.

There are per-request ingest charges when using PUT, COPY, or lifecycle rules to move data into any S3 storage class. Consider the ingest or transition cost before moving objects into any storage class. Estimate your costs using the AWS Pricing Calculator. To find the best S3 storage class for your workload, learn more here.

Region: US East (N. Virginia)

	Storage pricing
S3 Standard - General purpose storage for any type of data, typically used for frequently accessed data	
First 50 TB / Month	$0.023 per GB
Next 450 TB / Month	$0.022 per GB
Over 500 TB / Month	$0.021 per GB
S3 Intelligent - Tiering* - Automatic cost savings for data with unknown or changing access patterns	
Monitoring and Automation, All Storage / Month (Objects > 128 KB)	$0.0025 per 1,000 objects
Frequent Access Tier, First 50 TB / Month	$0.023 per GB

Amazon S3 pricing

- Pay only for what you use, including –
  - GBs per month
  - Transfer OUT to other Regions
  - PUT, COPY, POST, LIST, and GET requests

- You do not pay for –
  - Transfers IN to Amazon S3
  - Transfers OUT from Amazon S3 to Amazon CloudFront or Amazon EC2 in the same Region

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon S3: Storage pricing (1 of 2)

To estimate Amazon S3 costs, consider the following:

1. Storage class type –
   - Standard storage is designed for:
     - 11 9s of durability
     - Four 9s of availability
   - S3 Standard-Infrequent Access (S-IA) is designed for:
     - 11 9s of durability
     - Three 9s of availability
2. Amount of storage –
   - The number and size of objects

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Amazon S3: Storage pricing (2 of 2)

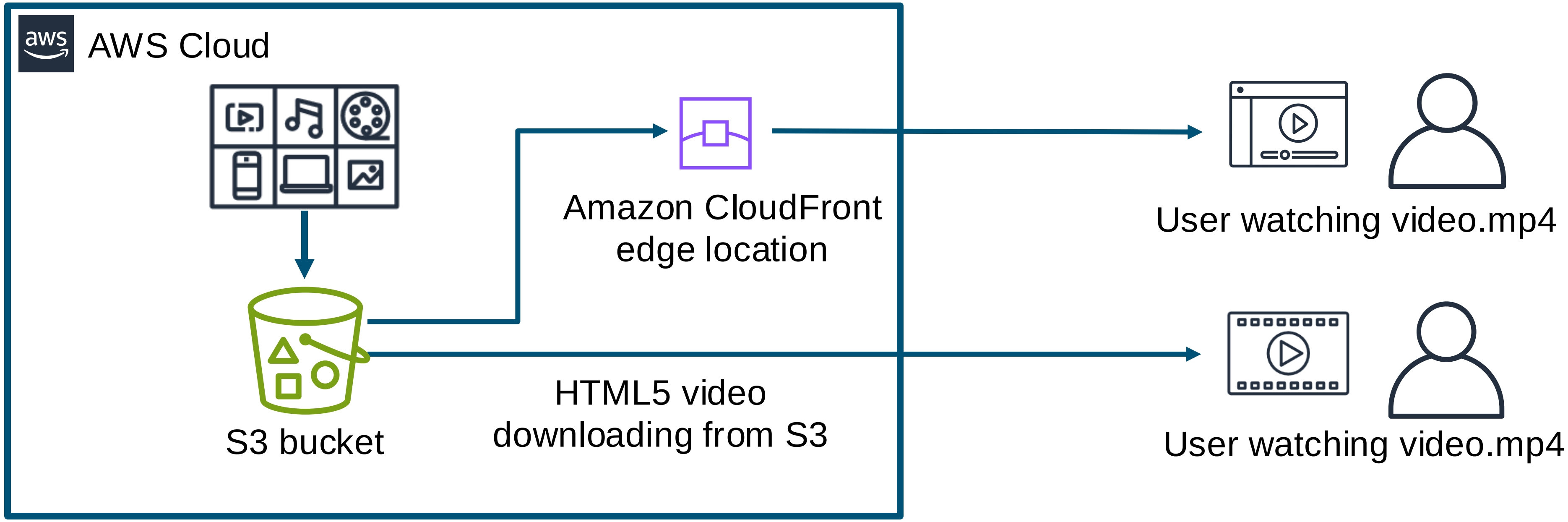
3. Requests –
    - The number and type of requests **(GET, PUT, COPY)**
    - Type of requests:
        - Different rates for GET requests than other requests.
4. Data transfer –
    - Pricing is based on the amount of data that is transferred out of the Amazon S3 Region
        - Data transfer in is free, but you incur charges for data that is transferred out.

© 2022, Amazon Web Services, Inc. or its affiliates. All rights reserved.

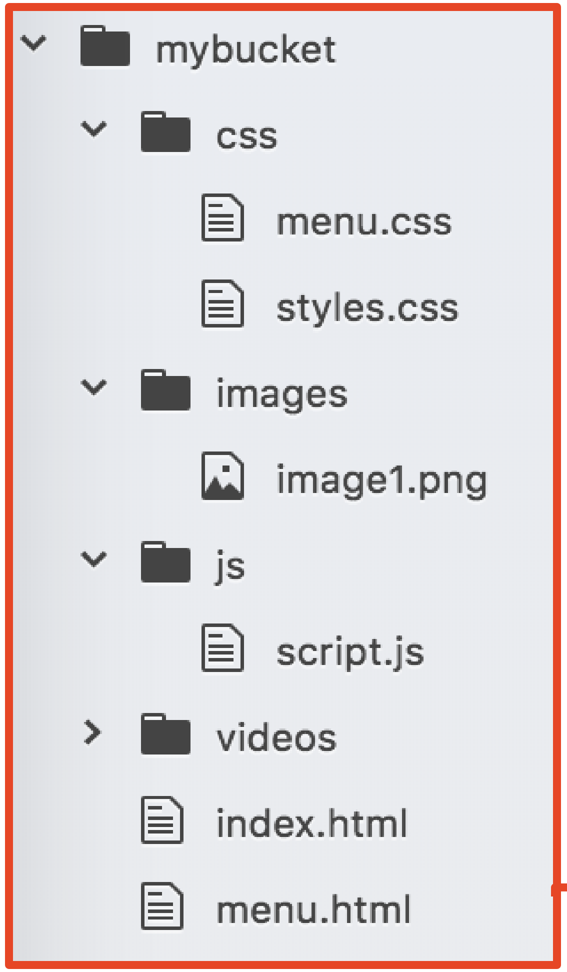
Common Use Cases

Use case: Media hosting

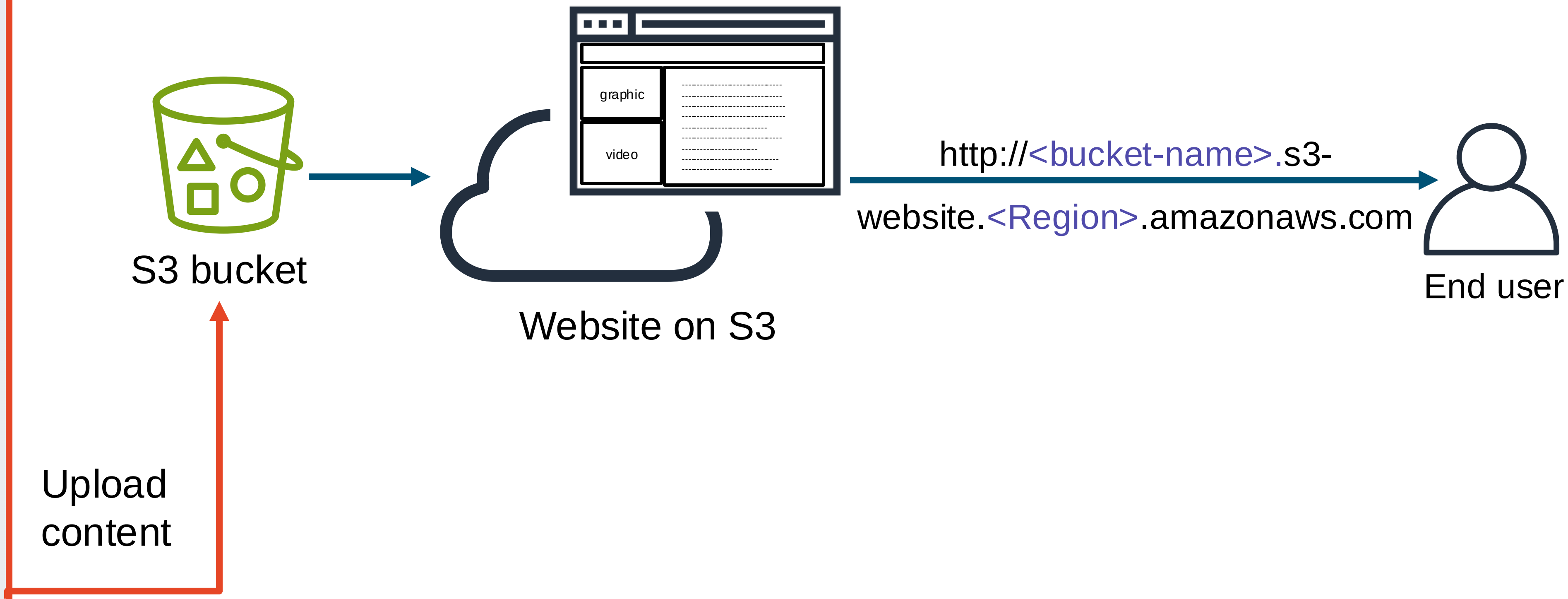
https://<bucket-name>.s3.amazonaws.com/video.mp4



Use case: Host static websites

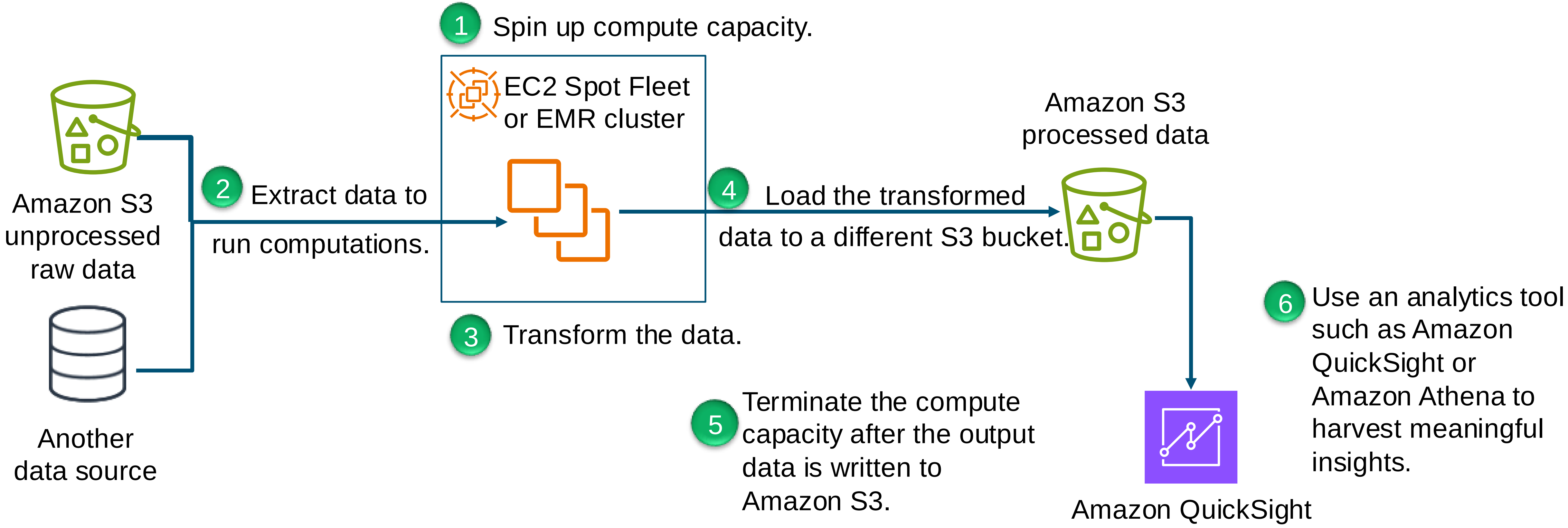


- mybucket
  - css
    - menu.css
    - styles.css
  - images
    - image1.png
  - js
    - script.js
  - videos
  - index.html
  - menu.html



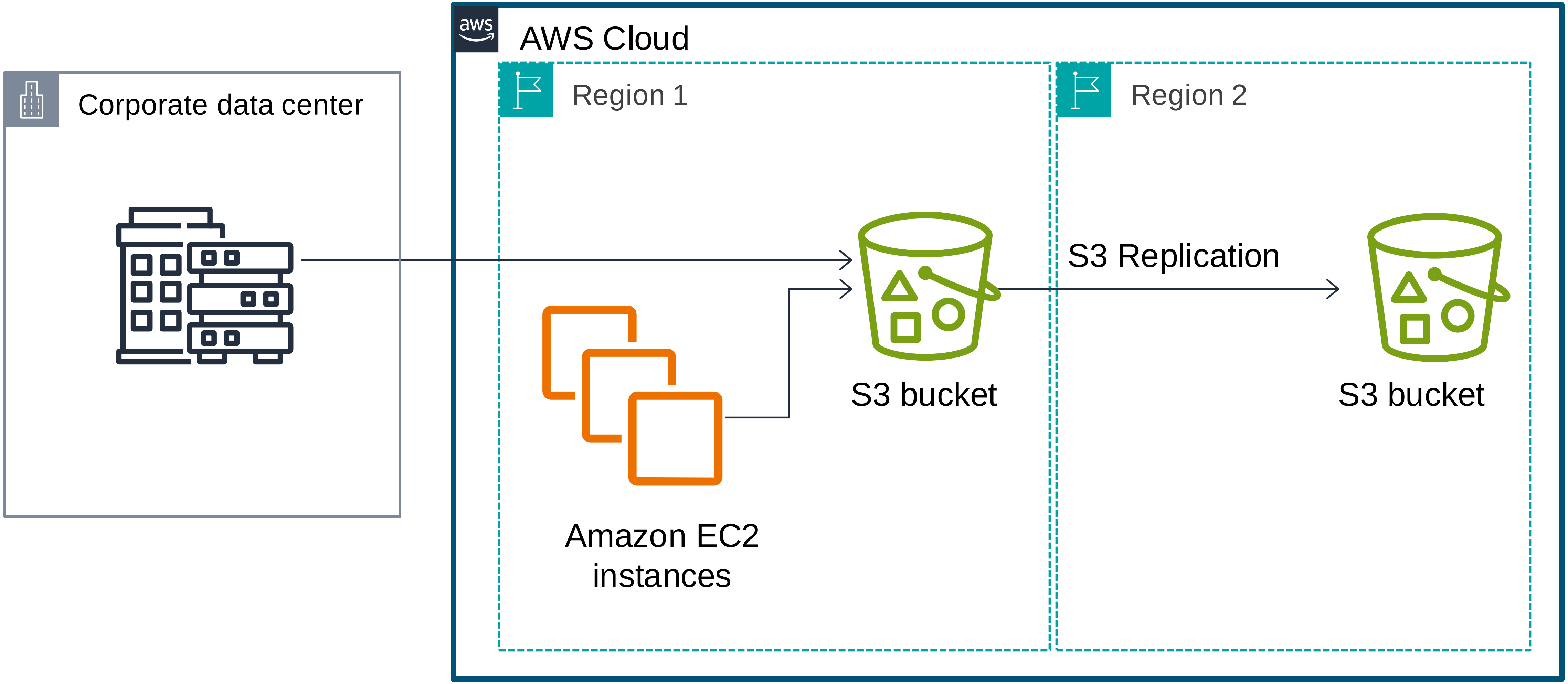
Use case: Data store for computation and analytics

Example data integration and preparation pattern



1 Spin up compute capacity.
EC2 Spot Fleet or EMR cluster
Amazon S3 unprocessed raw data
Another data source
2 Extract data to run computations.
3 Transform the data.
4 Load the transformed data to a different S3 bucket.
Amazon S3 processed data
5 Terminate the compute capacity after the output data is written to Amazon S3.
Amazon QuickSight
6 Use an analytics tool such as Amazon QuickSight or Amazon Athena to harvest meaningful insights.

Use case: Back up and archive critical data



S3 Storage Options

Object storage classes

General purpose	Intelligent tiering	Infrequent access	Archive
S3 Standard	S3 Intelligent-Tiering	S3 Standard-IA	S3 Glacier Instant Retrieval
			S3 Glacier Flexible Retrieval
		S3 One Zone-IA	S3 Glacier Deep Archive
			S3 on Outposts

S3 storage classes breakdown

	S3 Standard	S3 Intelligent-Tiering	S3 Standard-IA	S3 One Zone-IA	S3 Glacier Instant Retrieval	S3 Glacier Flexible Retrieval	S3 Glacier Deep Archive
Availability Zones	≥3	≥3	≥3	1	≥3	≥3	≥3
Minimum capacity charge for each object	N/A	N/A	128 KB	128 KB	128 KB	N/A	N/A
Minimum storage duration charge	N/A	N/A	30 days	30 days	90 days	90 days	180 days
Retrieval charge	N/A	N/A	Per GB retrieved	Per GB retrieved	Per GB retrieved	Per GB retrieved	Per GB retrieved

Configuring an Amazon S3 Lifecycle

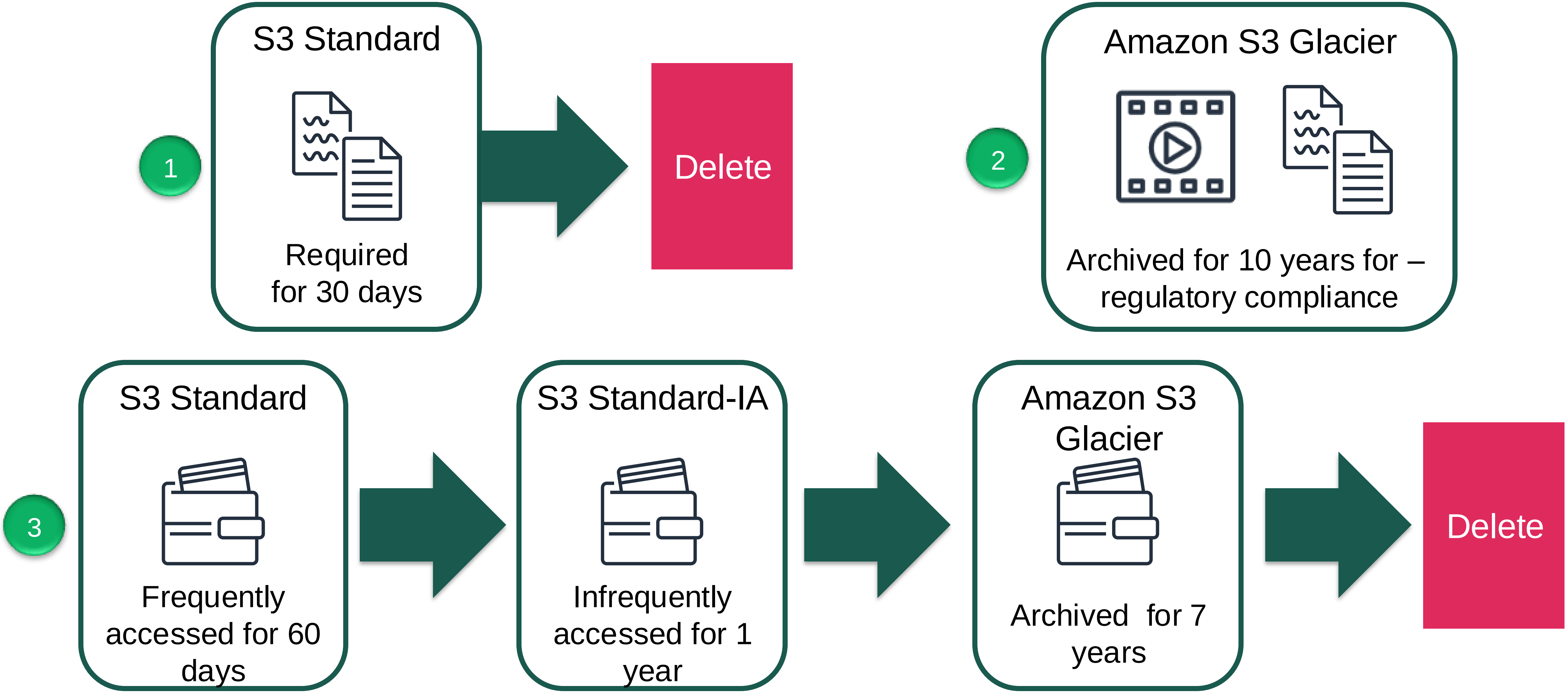
Amazon S3 lifecycle configurations are a set of rules that define actions that Amazon S3 applies to a group of objects.

- Transition actions transition to another storage class.
- Expiration actions define when objects expire.

Set an S3 lifecycle policy.

Data will automatically transfer to a different storage class without any changes to your application.

Amazon S3 lifecycle examples



1

S3 Standard

Required for 30 days

Delete

2

Amazon S3 Glacier

Archived for 10 years for – regulatory compliance

3

S3 Standard

Frequently accessed for 60 days

S3 Standard-IA

Infrequently accessed for 1 year

Amazon S3 Glacier

Archived for 7 years

Delete

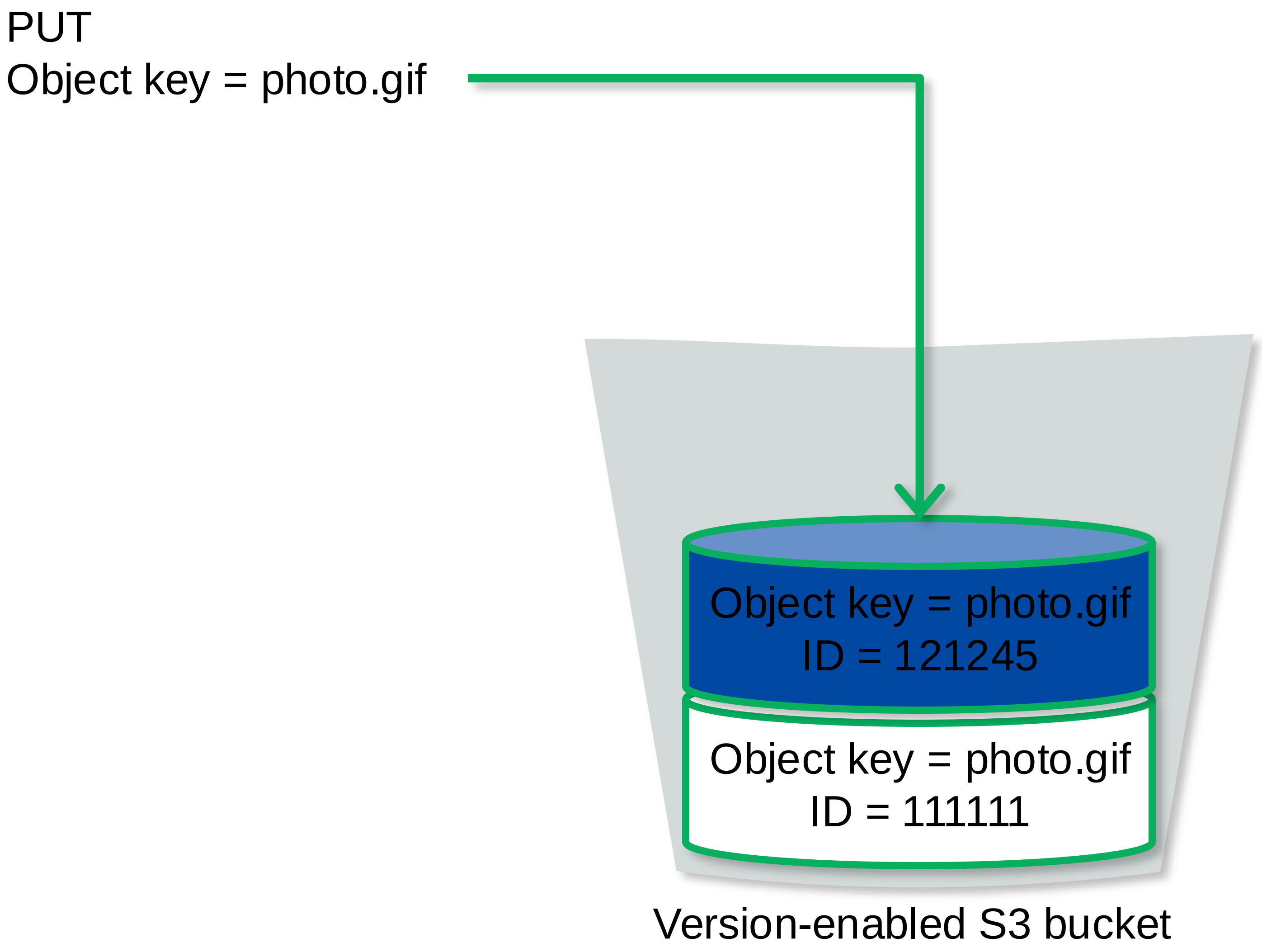
S3 Versioning

Amazon S3 versioning

Amazon S3 versioning protects objects from *accidental overwrites* and *deletes*.

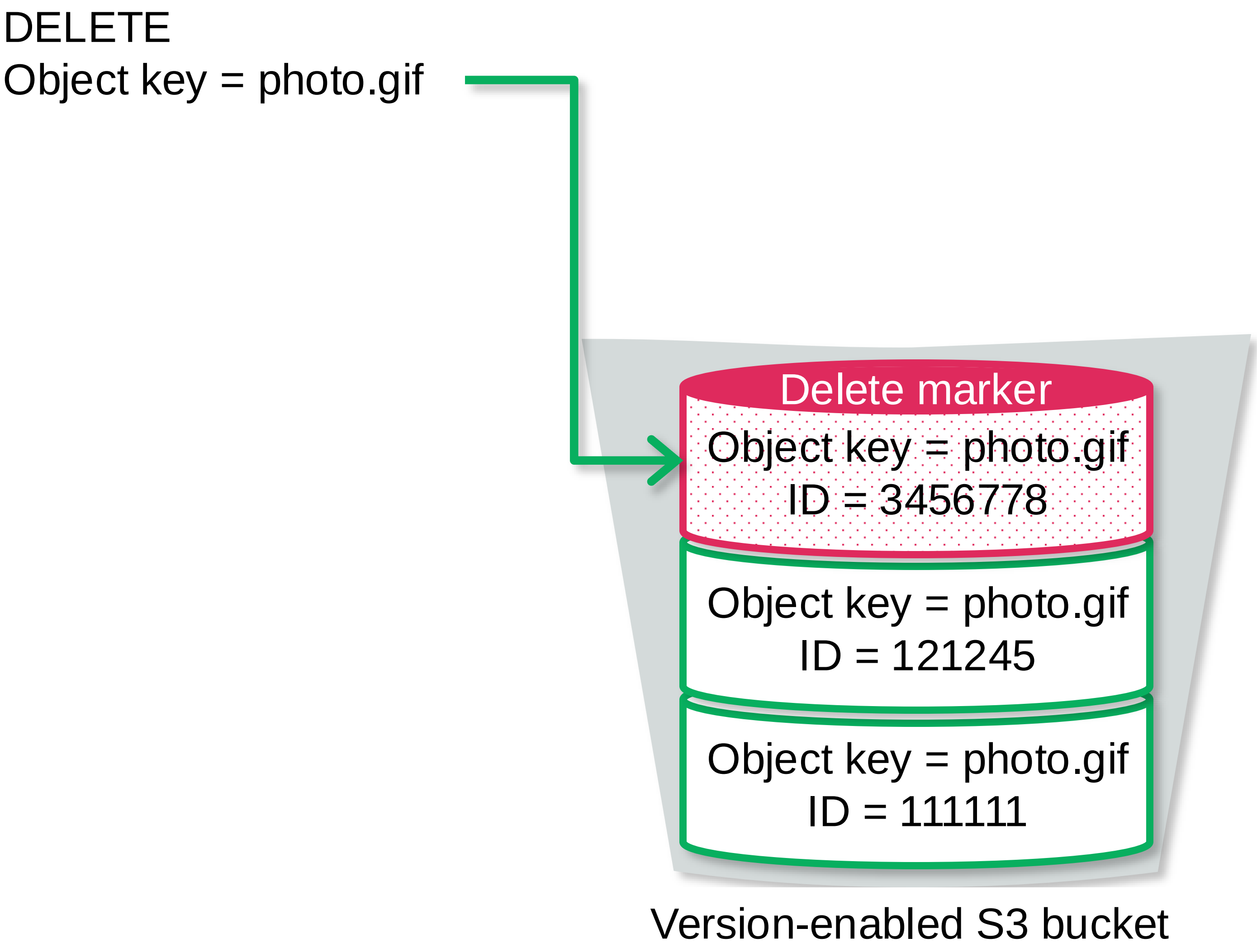
Action	Versioning Enabled	Versioning Disabled or Versioning Suspended
Upload an object with the same key	Creates a new object with a different version ID, and both are retrievable by the version ID.	Overwrites the original object, and the previous object is no longer retrievable.
Delete	Adds a delete marker, but the object is still retrievable by the version ID.	Deletes the object, and it is no longer retrievable.

Adding an object in a versioning-enabled bucket



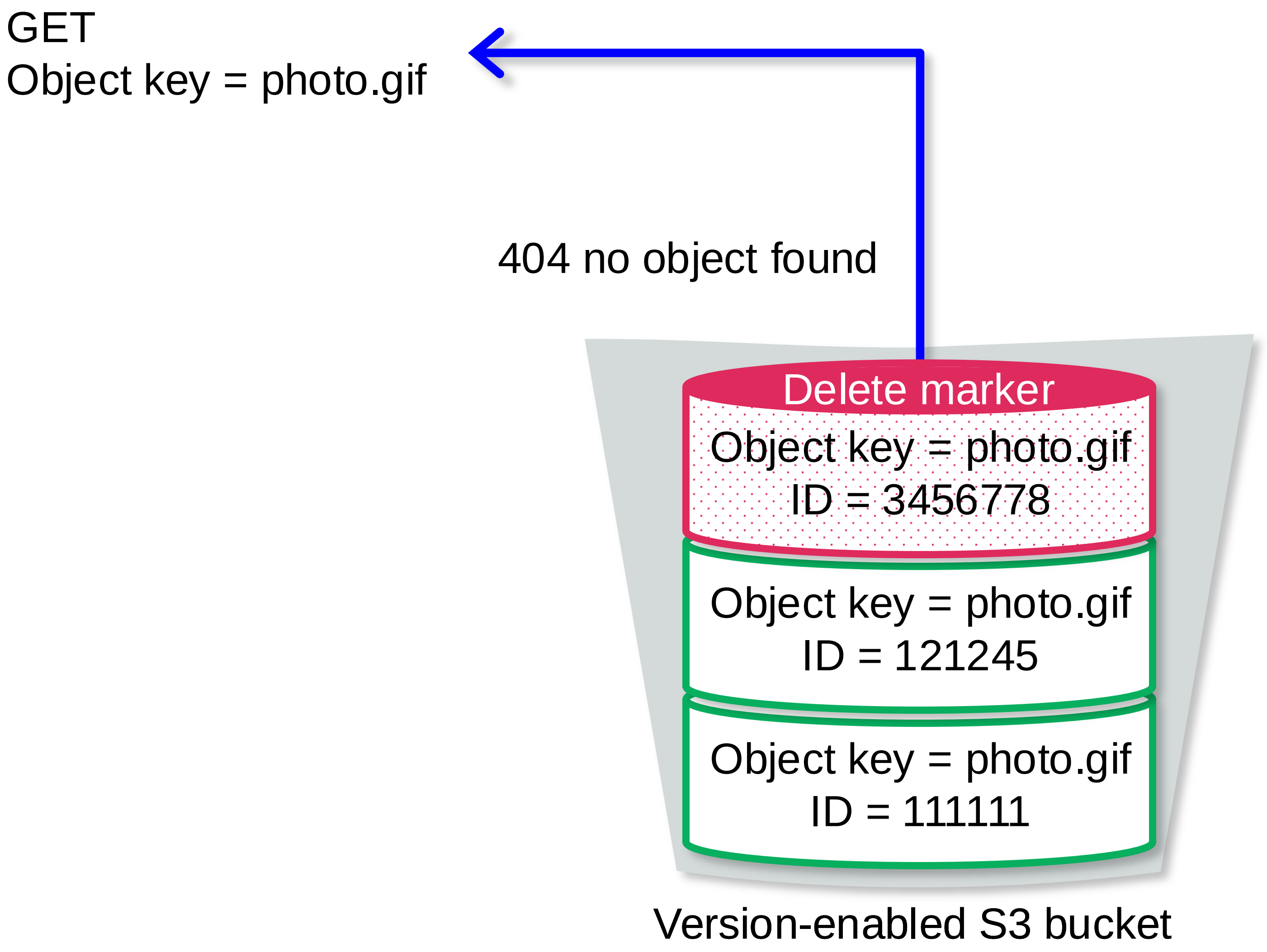
- Amazon S3 generates a new version ID and adds this newer version of the object to the bucket.
- The original version remains in the bucket.

Deleting an object in a version-enabled bucket



When a request is made to delete an object in a version-enabled bucket, all versions remain in the bucket, but Amazon S3 inserts a delete marker.

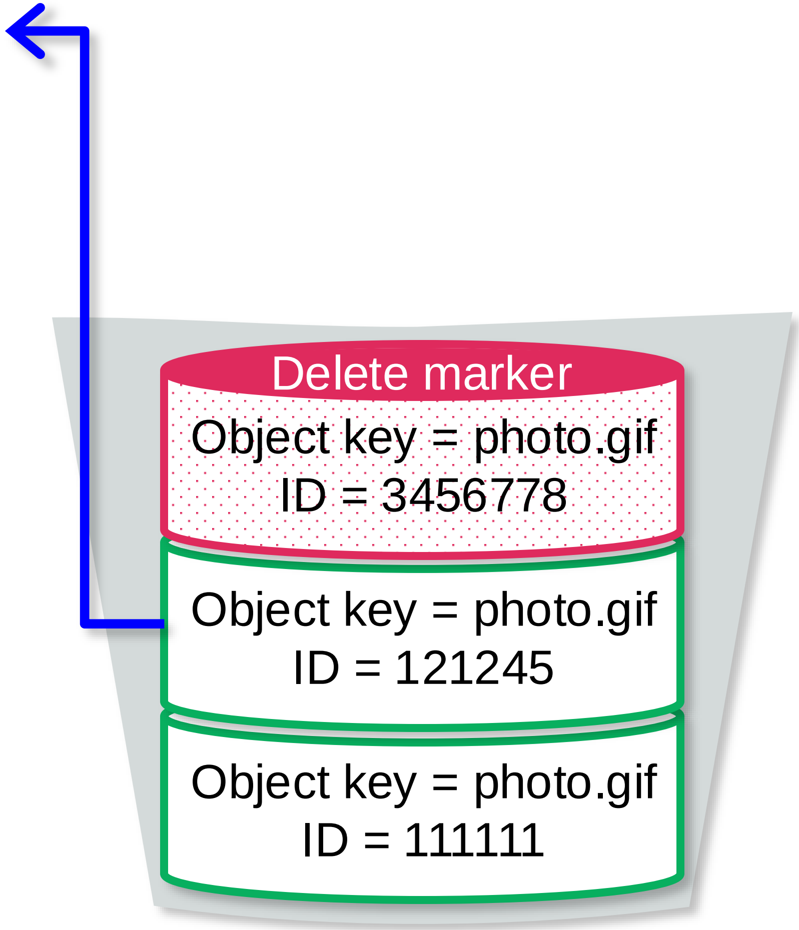
Retrieving the most recently stored version



- Requests for an object key return the most recent version.
- If the most recent version is a delete marker, the request is not successful.

Retrieving an object with its specific ID

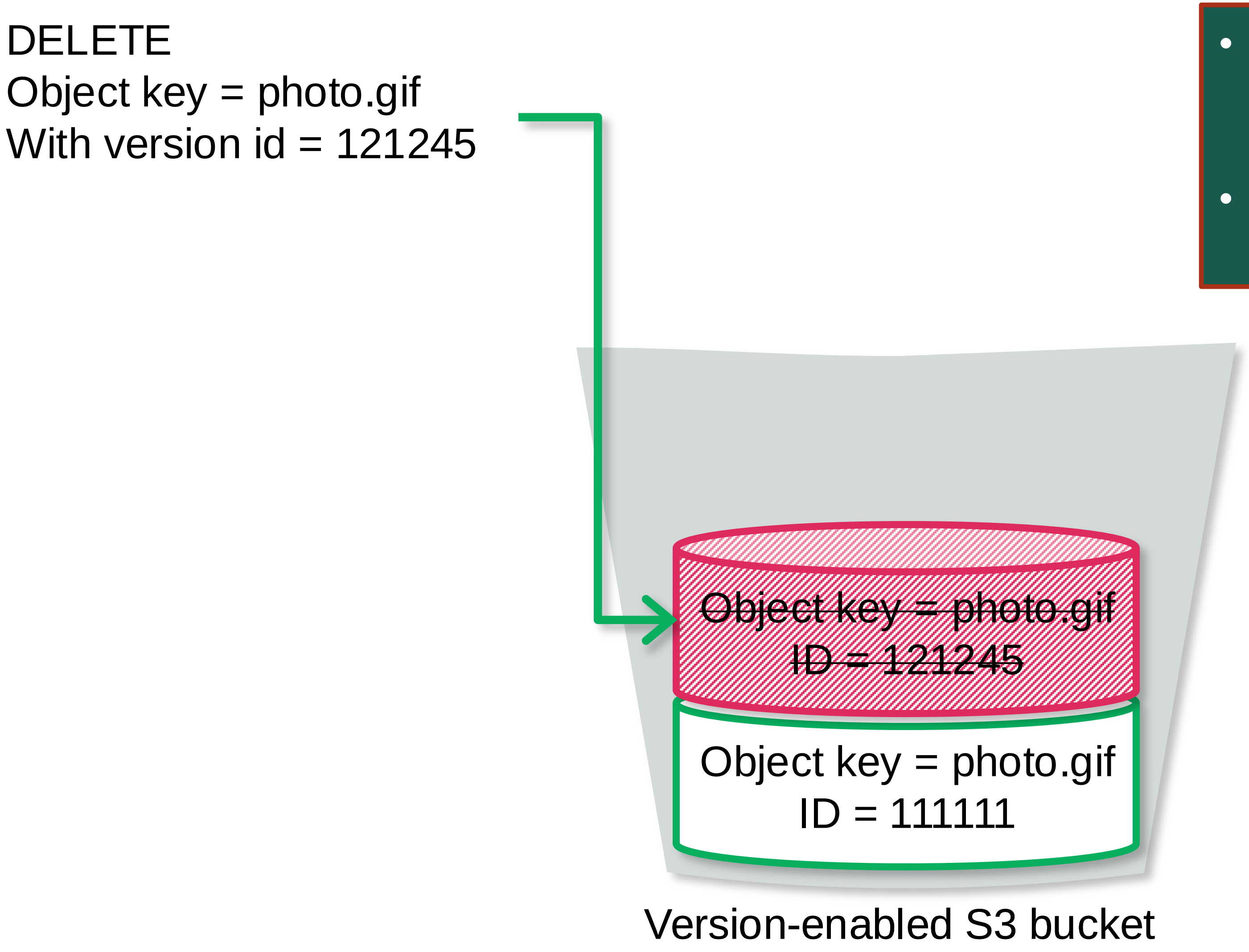
GET
Object key = photo.gif with
version ID = 121245



Version-enabled S3 bucket

Requests for an object with its version ID will successfully return that version of the object.

Permanently delete an object



DELETE
Object key = photo.gif
With version id = 121245

~~Object key = photo.gif~~
~~ID = 121245~~

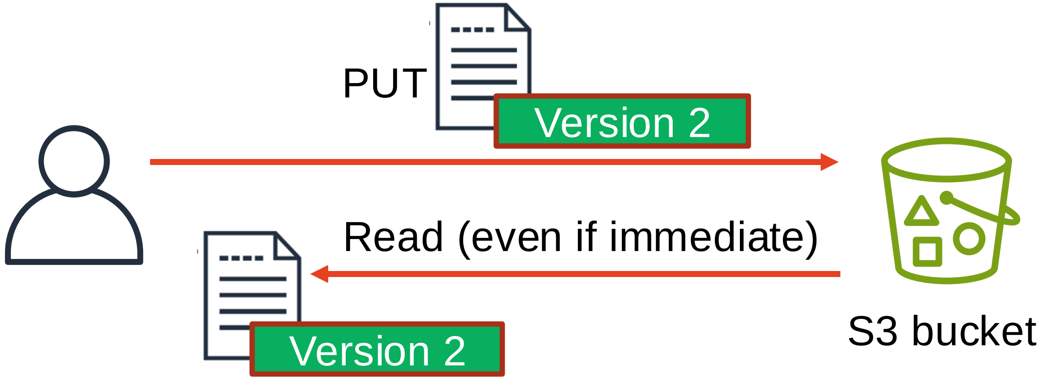
Object key = photo.gif
ID = 111111

Version-enabled S3 bucket

- Owners of the bucket can permanently delete an object by using delete with the version ID.
- In this case, no delete marker is added, and the specified version is not recoverable.

Amazon S3 data consistency model

- Is consistent for all new and existing objects in all Regions
- Provides read-after-write consistency for all GET, LIST, PUT, and DELETE operations on objects in S3 buckets
- Offers an advantage for big data workloads
- Simplifies the migration of on-premises analytics workloads



S3 redundancy vs data versioning

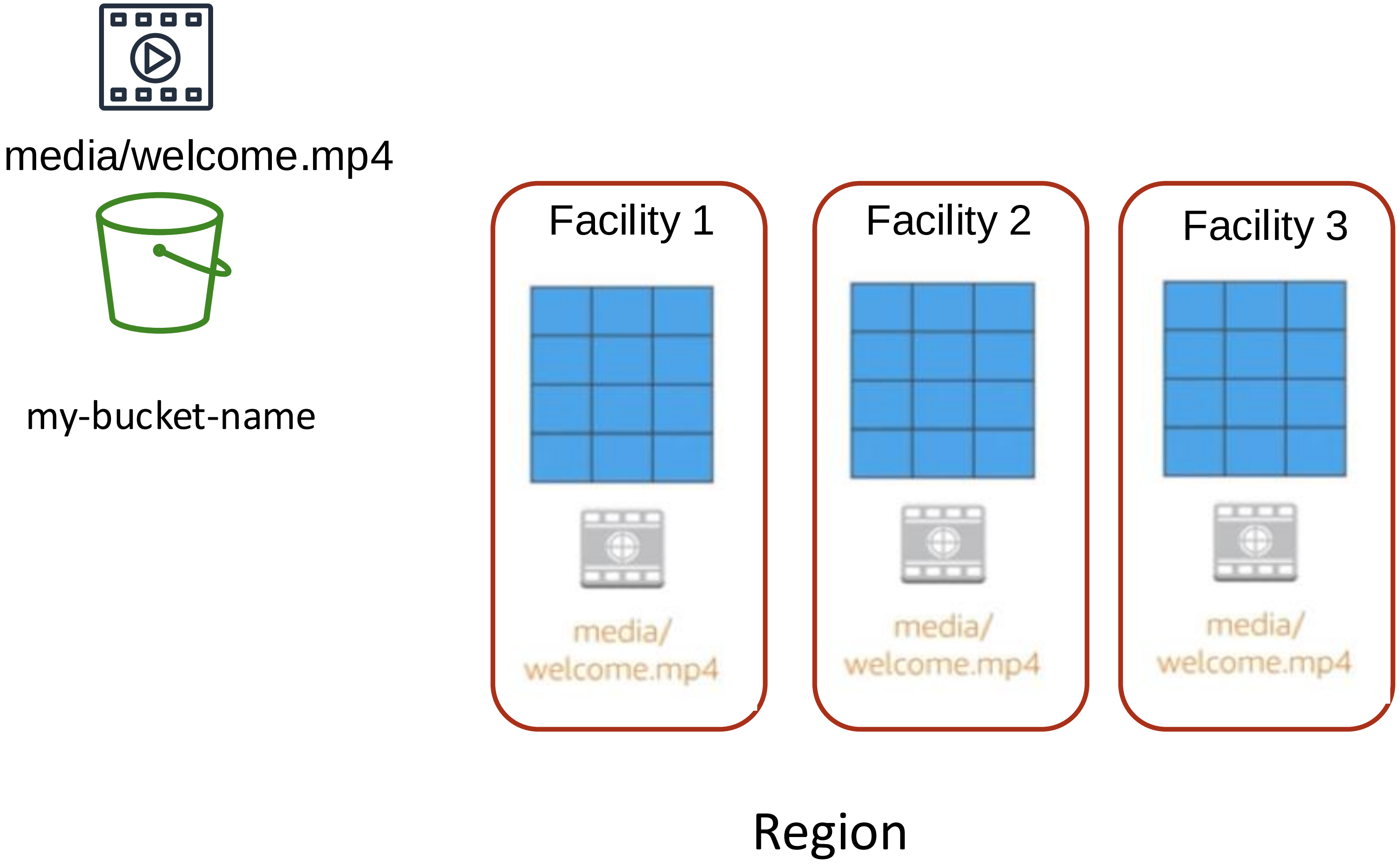
S3 Redundancy

- **Purpose:**
  Ensures high availability and durability of data.
- **How It Works:**
  - Automatically stores data across multiple devices and Availability Zones.
  - Protects against hardware failures and data center outages.
- **Key Benefit:**
  - Provides fault tolerance and continuous access to data.
- **Extra cost**: none; this is the default feature

S3 Versioning

- **Purpose:**
  Protects against accidental deletions and overwrites.
- **How It Works:**
  - Maintains multiple versions of an object in the same bucket
  - Enables restoration of previous versions if needed.
- **Key Benefit:**
  Offers a robust audit trail and easy recovery of data modifications.
- **Extra cost**: may incur extra storage cos

S3 Redundancy and Versioning



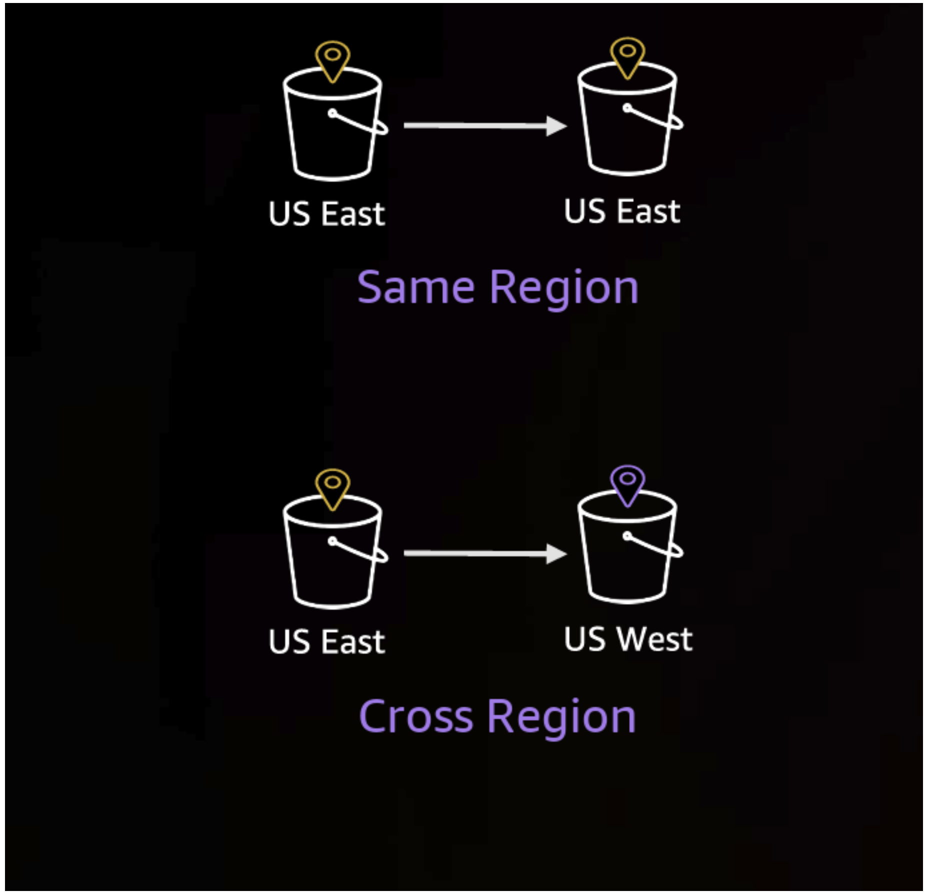
- When versioning is enabled, each version will be redundantly stored in multiple facilities by default

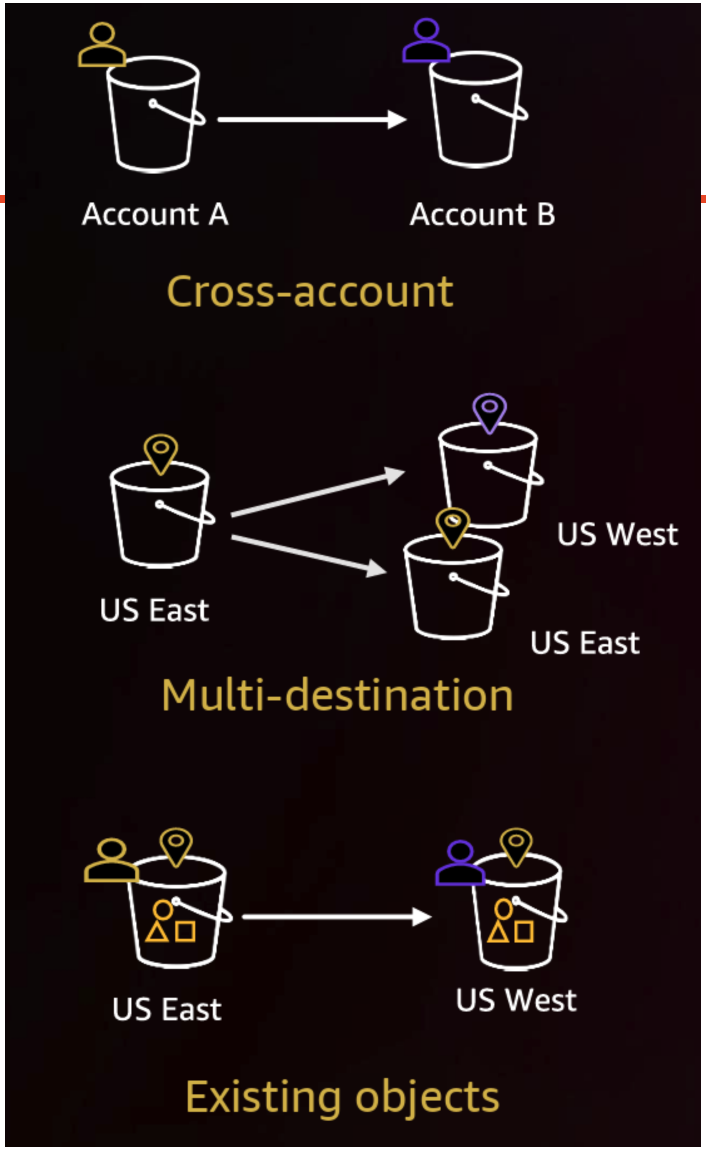
S3 Replication

What is S3 Replication?

- S3 replication is a feature that automatically copy objects from one bucket (*source bucket*) to another bucket(s) (*destination bucket(s)*)
- S3 replication requires that versioning to be enabled in both the source and the destination buckets
- Why do users replicate data?
  - Additional disaster recovery (to measures like data redundancy)
  - Satisfy compliance
  - Latency improvement

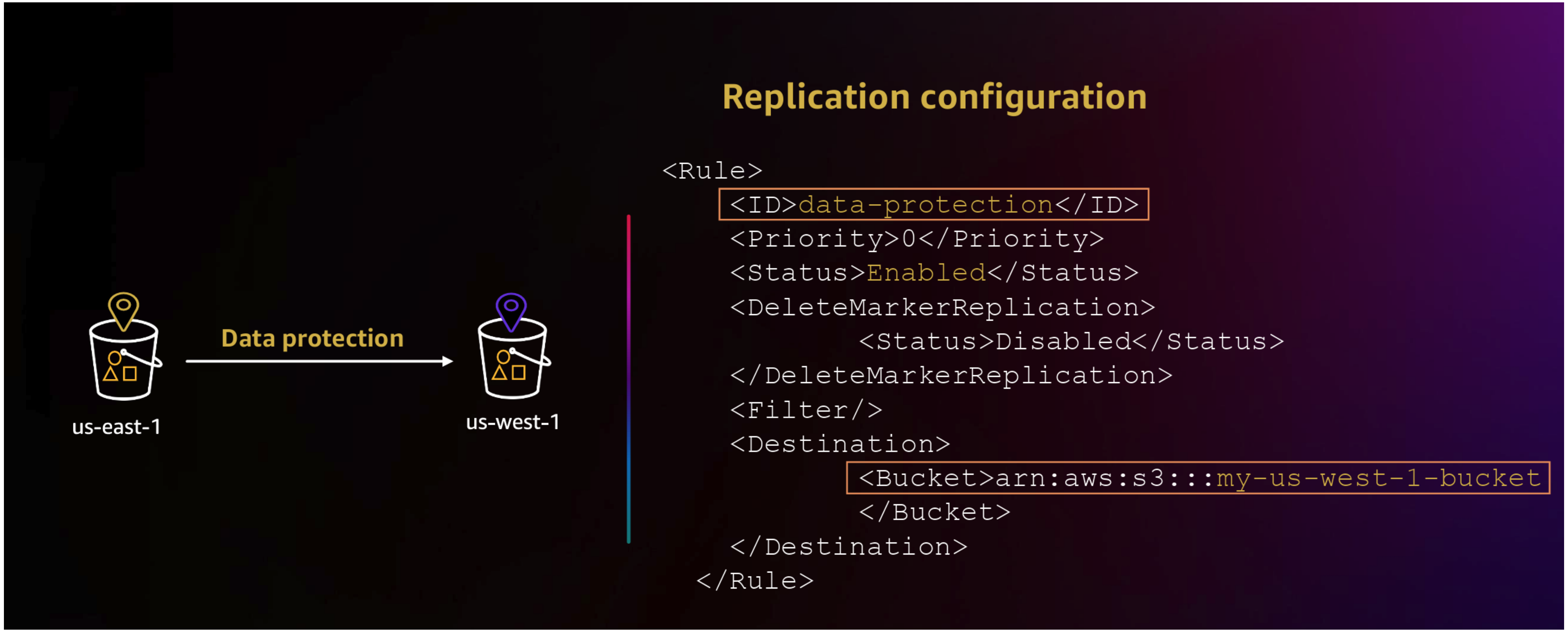
Various replication features





AWS re:Invent 2022 - Best practices for Amazon S3 replication, featuring HBO Max (STG336)

A simple replication case

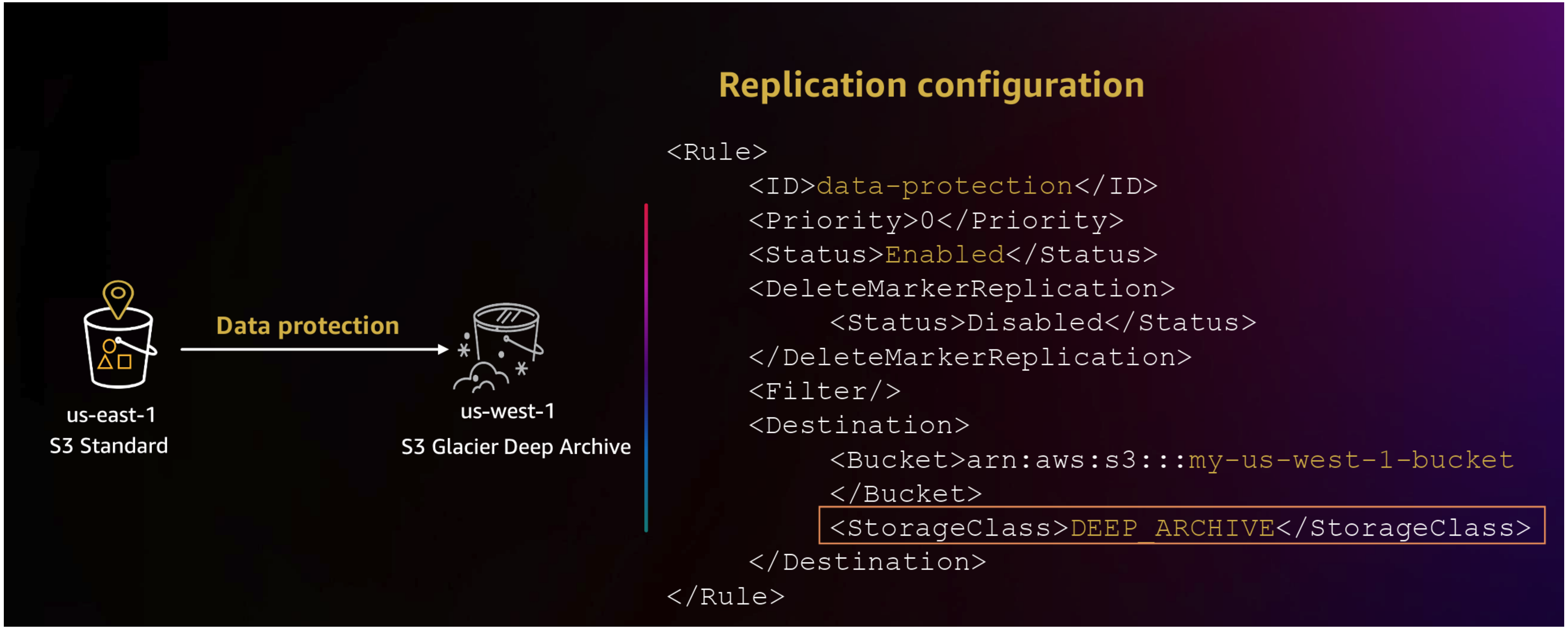


Replication configuration

```
<Rule>
    <ID>data-protection</ID>
    <Priority>0</Priority>
    <Status>Enabled</Status>
    <DeleteMarkerReplication>
            <Status>Disabled</Status>
    </DeleteMarkerReplication>
    <Filter/>
    <Destination>
            <Bucket>arn:aws:s3:::my-us-west-1-bucket
            </Bucket>
    </Destination>
  </Rule>
```

AWS re:Invent 2022 - Best practices for Amazon S3 replication, featuring HBO Max (STG336)

Replication to achieve storage cost efficiency



us-east-1
S3 Standard

Data protection

us-west-1
S3 Glacier Deep Archive

Replication configuration

```
<Rule>
     <ID>data-protection</ID>
     <Priority>0</Priority>
     <Status>Enabled</Status>
     <DeleteMarkerReplication>
          <Status>Disabled</Status>
     </DeleteMarkerReplication>
     <Filter/>
     <Destination>
          <Bucket>arn:aws:s3:::my-us-west-1-bucket
          </Bucket>
          <StorageClass>DEEP_ARCHIVE</StorageClass>
     </Destination>
</Rule>
```

AWS re:Invent 2022 - Best practices for Amazon S3 replication, featuring HBO Max (STG336)

How delete operations affect replication

- If you make a DELETE request *without specifying an object version ID*, Amazon S3 adds a delete marker. Amazon S3 deals with the delete marker as follows:
  - By default, Amazon S3 adds a delete marker in the source bucket *only*.
  - The delete marker is NOT copied to the destination bucket by default.
  - This could cause some consistency issue: an object is deleted in the source but not the destination
  - We can have ***delete marker replication*** enabled, these markers are copied to the destination buckets, and Amazon S3 behaves as if the object was deleted in both the source and destination buckets.

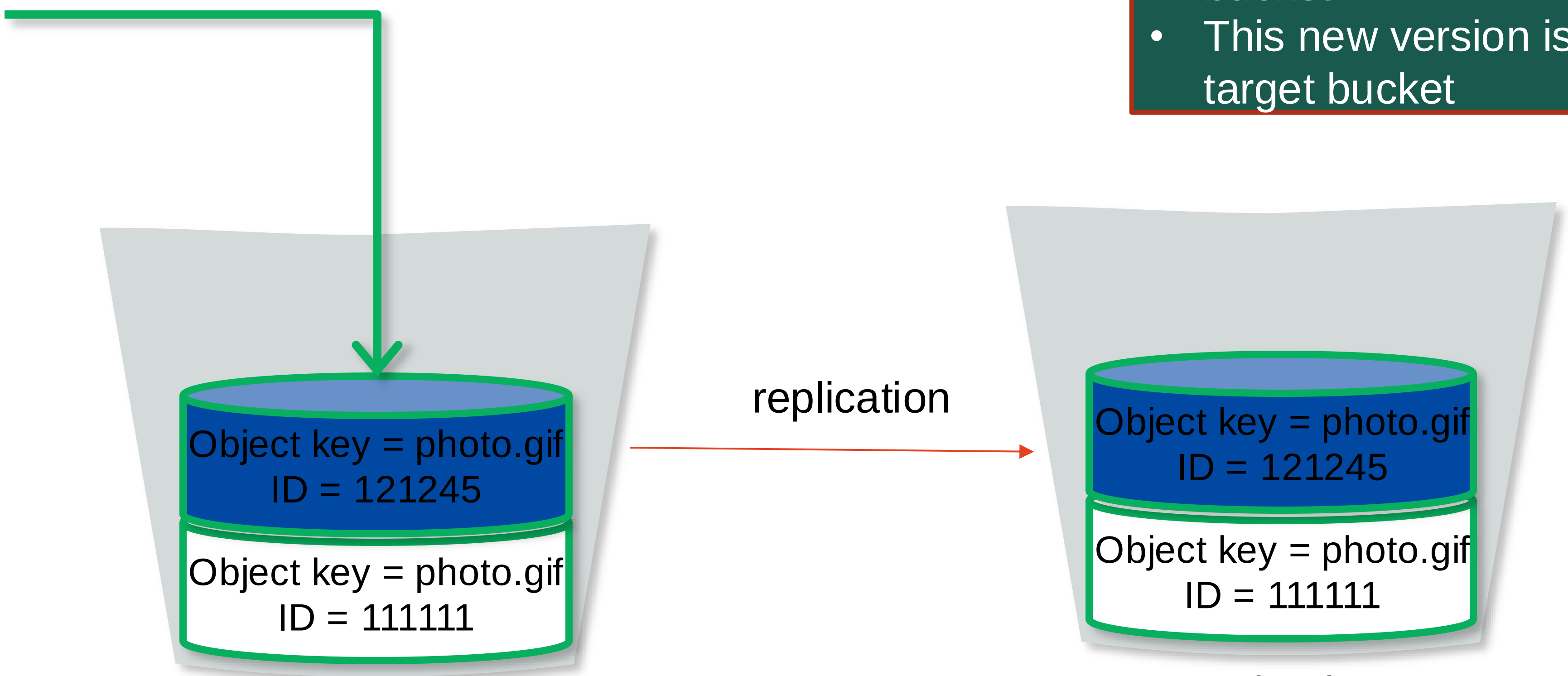
How delete operations affect replication (cont’d)

- If you specifying an object version ID to delete in a DELET request
  - Amazon S3 delete that object version in the source bucket only
  - The delete operation is NOT replicated to destination bucket
  - This could cause some consistency issue: an object version is deleted in the source but not the destination
  - Users will need to manually delete the version in the target bucket

S3 Replication and Versioning

- Amazon S3 generates a new version ID and adds this newer version of the object to the bucket.
- The original version remains in the bucket.
- This new version is replicated to the target bucket

PUT
Object key = photo.gif



Object key = photo.gif
ID = 121245

Object key = photo.gif
ID = 111111

S3 source bucket

replication

Object key = photo.gif
ID = 121245

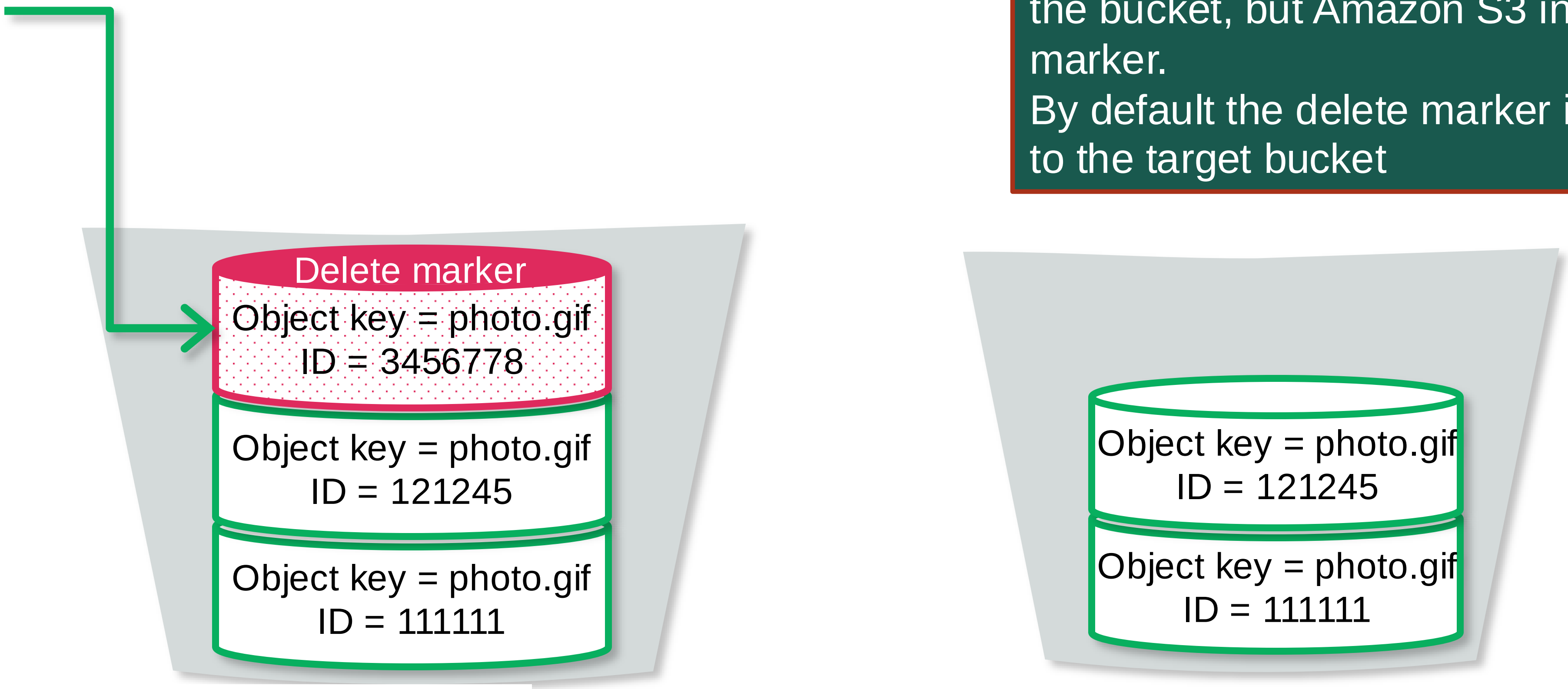
Object key = photo.gif
ID = 111111

S3 target bucket

Default behaviour in replicated bucket

DELETE
Object key = photo.gif

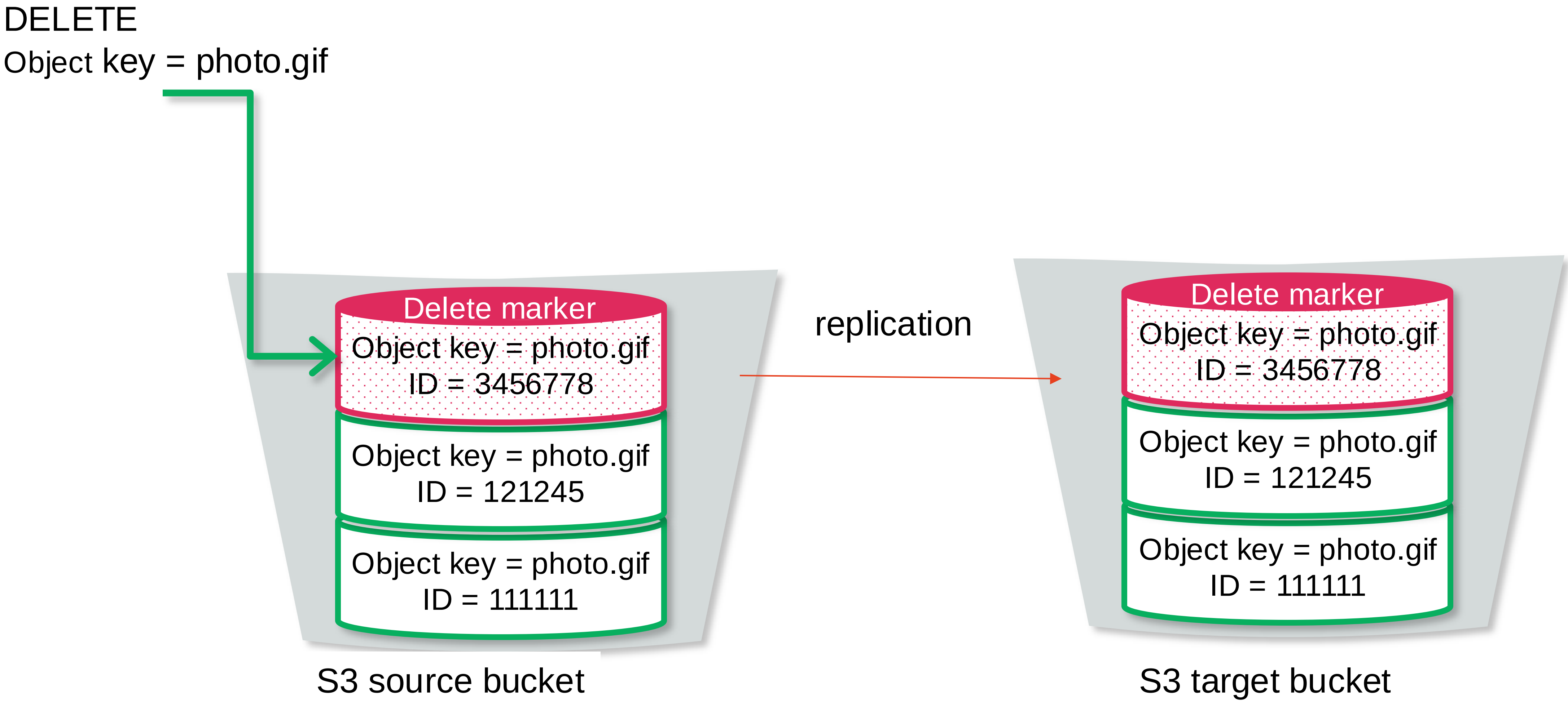
When a request is made to delete an object in the source bucket, all versions remain in the bucket, but Amazon S3 inserts a delete marker.
By default the delete marker is not replicated to the target bucket



S3 source bucket

S3 target bucket

Delete marker replication enabled



DELETE
Object key = photo.gif

Delete marker
Object key = photo.gif
ID = 3456778

Object key = photo.gif
ID = 121245

Object key = photo.gif
ID = 111111

S3 source bucket

replication

Delete marker
Object key = photo.gif
ID = 3456778

Object key = photo.gif
ID = 121245

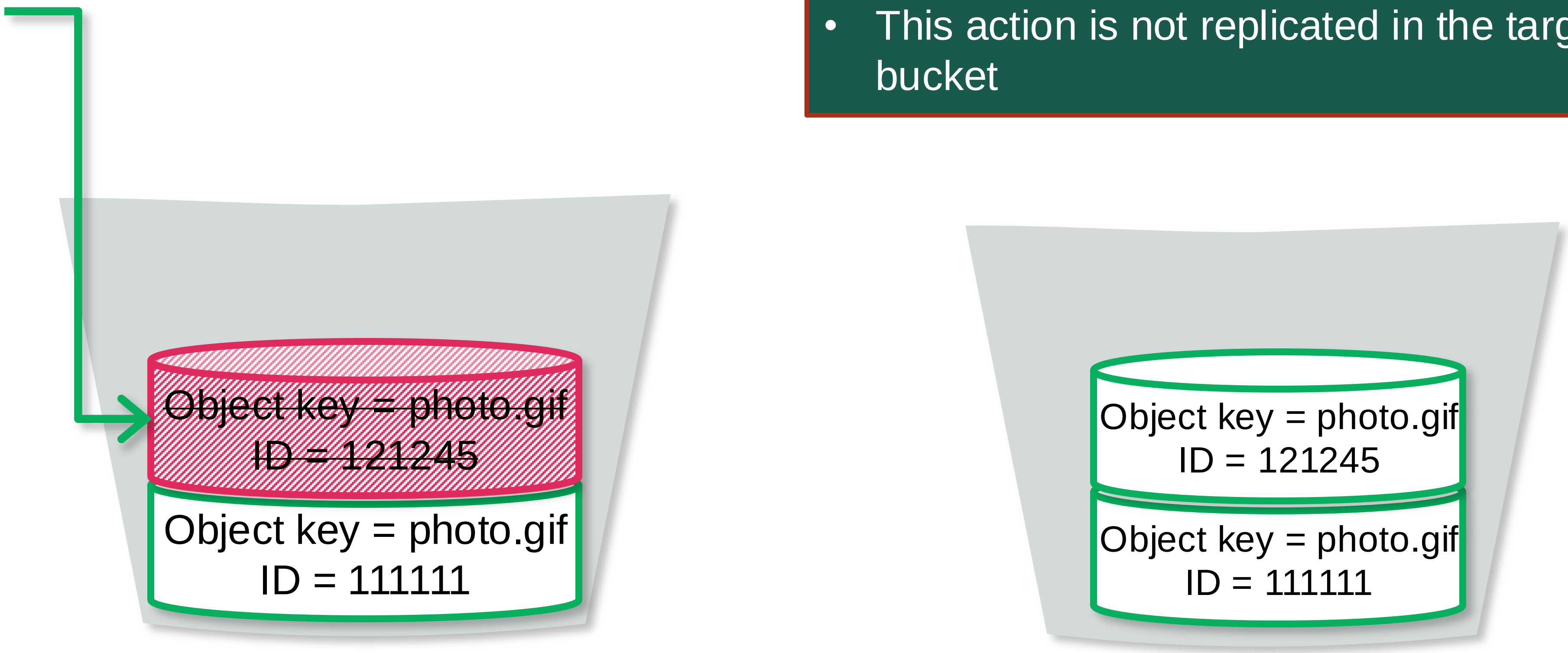
Object key = photo.gif
ID = 111111

S3 target bucket

Delete a specific version

DELETE
Object key = photo.gif
With version id = 121245

- When the delete operation specifies a version ID.
- no delete marker is added, and the specified version is will be deleted in the source bucket
- This action is not replicated in the target bucket



~~Object key = photo.gif~~
~~ID = 121245~~

Object key = photo.gif
ID = 111111

S3 source bucket

Object key = photo.gif
ID = 121245

Object key = photo.gif
ID = 111111

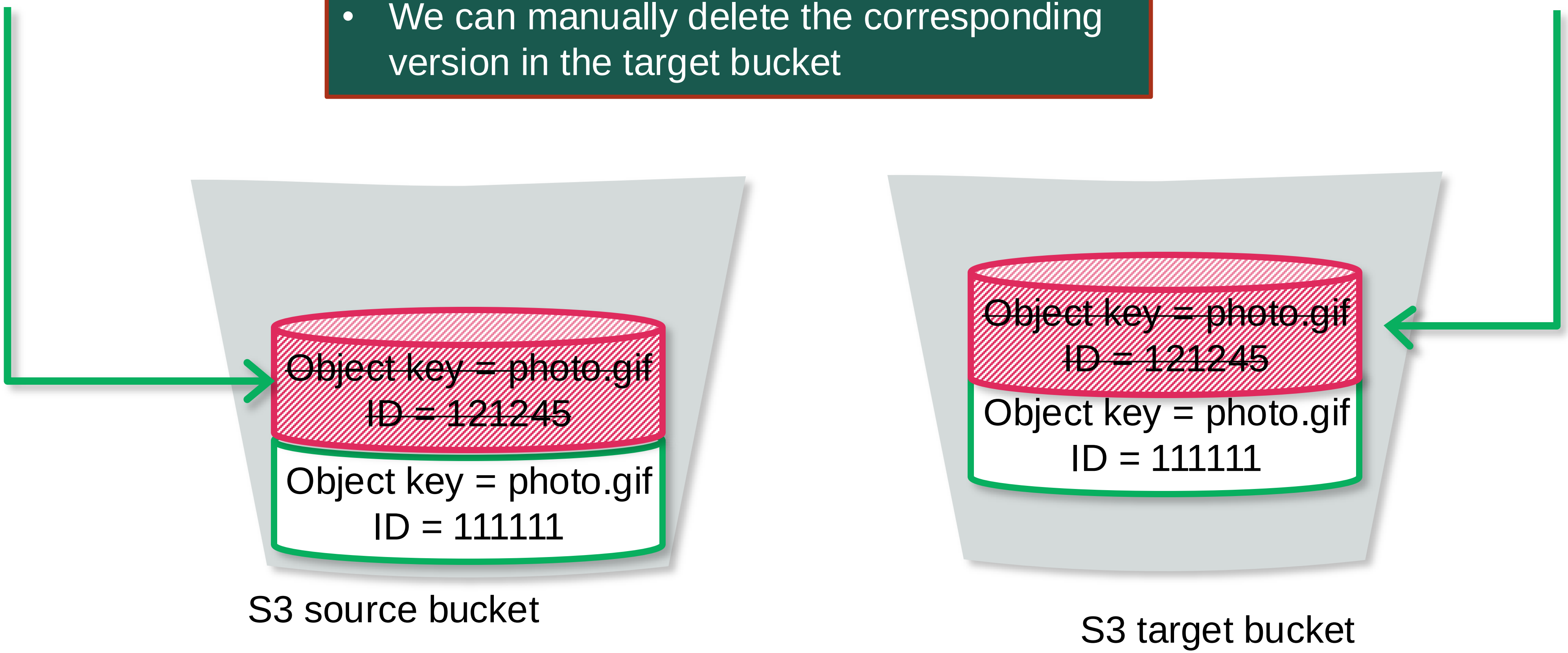
S3 target bucket

Delete a specific version

DELETE
Object key = photo.gif
With version id = 121245

- We can manually delete the corresponding version in the target bucket

DELETE
Object key = photo.gif
With version id = 121245



References

- Michael Wittig, Andreas Wittig, **Amazon Web Services in Action, Third Edition**
  - Chapter 7
- AWS Academy Cloud Architecting
  - Module 4 slides
- AWS documentation
  - https://docs.aws.amazon.com/AmazonS3/latest/userguide/replication.html

THE UNIVERSITY OF
SYDNEY