# BackUp Strategies


## Amazon Web Services

# Backup Strategies

Backup strategies are important to businesses of all sizes. Understanding available options and their differences is no easy task. To help narrow down the search, this course focuses on Amazon S3 and Amazon Glacier for backup and archival solutions.

What this course covers:

- Traditional backup and archival options
- Amazon S3 and Glacier as alternatives
- Data transfer options
- Security for data at rest and in transit

# What are backup strategies?

Backup strategies are a defense against data loss.

They are made up of multiple different components, like:

- A pre-defined schedule for backing up information
- The type of backup to be performed – full backup? Incremental? Automated?
- What needs to be backed up – the entire network? Specific drives or data?
- Backup and archive locations
- …and more

# Onsite Backups

Onsite backups generally refers to storing data on local storage devices like hard disk drives, magnetic tapes, and other options.

Onsite storage has a number of benefits:

- Data can be quickly accessed
- It gives complete control over the hardware and storage
- No internet access is needed (great for security and bandwidth concerns)

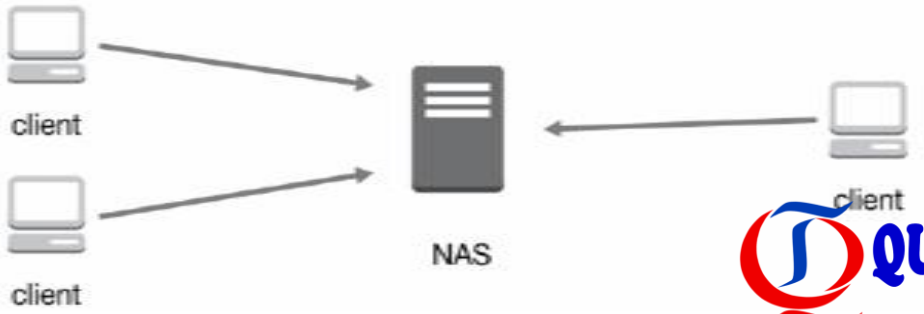
Onsite storage has disadvantages:

- It ties up resources because of hardware, facility, and human resource costs
- Having the data in one location is dangerous in the case of a natural disaster like fire, flooding, or earthquakes
- Storage devices can be stolen or accessed, so physical security is very important

# Onsite Backup Options - NAS

NAS stands for Network Attached Storage and it is a central storage hub which can be used for backing up data.

- A NAS is connected to the local network, making it accessible to anyone connected to that network
- Multiple users can access a NAS at the same time
- They are scalable (you can RAID drives)
- Advanced devices come with advanced features (clustering, replication, and data consolidation, for example)



# Onsite Backup Options - Tape

Tape is another storage option which can provide more durable storage than disk drives.

- Fewer moving and fragile parts
- Less susceptible to damage from dust, humidity, and other potential damage
- Option often used for long-term backups and archives
- Not as fast for random access, but fast for sequential access
- Can be used to complement disk drives as part of the backup strategy

## Offsite Backups

Offsite storage has a number of benefits:

- Can access data from multiple locations via the Internet or other means
- Protects against any event happening onsite (like a natural disaster for example)
- Dedicated offsite locations are built for storage (climate control, security, compliance, etc...)

Offsite storage has disadvantages:

- Internet access requirements (may require a lot of bandwidth and fast internet)
- Can have higher cost per GB of data stored
- Can take longer to recover data

# Offsite Backups

There are different models when dealing with offsite storage:

- Rent equipment and physical space
- Own equipment and rent physical space
- Storage as a service (pay per GB for example)

## Onsite-Offsite Backups

Both options can be used to counteract some of their disadvantages.

For example: By having data offsite in addition to onsite, we can protect against disastrous events. The data is replicated in multiple locations, so if a fire or other event occurred, we could still recover the data from the other geographically separated location.

Having data onsite can also speed up recovery times and transfer times, since we wouldn't have to transfer massive amounts of data through the open Internet or through dedicated connections in order to recover data in the event of an issue.

## Cloud as an Alternative - Cost

The cloud can be used as an alternative to avoid many of the upfront costs associated with purchasing hardware, hiring experts, renting physical space, etc...

We can oftentimes pay for what we use and when we use it, instead of having to over-provision for future needs.

## Cloud as an Alternative - Complexity

The cloud reduces complexity:

- No need to worry about hardware or physical space
- No need to find "IBM tape experts" – we can use SDKs and APIs
- No need to architect durability, reliability, and availability – Amazon does it

# Cloud as an Alternative - Scalability

With the cloud, resources can be provisioned on demand without usually having to purchase capacity upfront (though there are exceptions to this).

Purchasing or renting resources in a datacenter requires buying more capacity than is needed in order to ensure scalability ahead of time.

Example: If we send a lot more data to Amazon S3, S3 will scale to handle it. If we had a limited amount of drives, we would have to purchase more and install those drives before they are needed, which requires planning and also costs more.

# Cloud as an Alternative – Issues

We usually require access to the Internet in order to backup and retrieve data. This means we have to rely on bandwidth, and we need to transfer our data over the open Internet.

How can we speed up large transfers, and how can we secure our data in transit and at rest? These are important questions we will look at throughout this course.

Using the cloud to backup and archive data means you are trusting the cloud provider with your data. Not only are you trusting that they have the proper physical security procedures in place, but also that they will deliver on their durability, reliability, and availability claims.

# What is Amazon S3?

Amazon S3 is a simple, durable, and massively scalable object storage service.

S3 can be used to store all kinds of different objects without worrying about hardware or upfront costs

This lesson will explain:

- The available storage classes
- Similarities and differences between those storage classes

# Amazon S3 Storage Classes - Standard

Standard storage is the default and the option that gives us:

Pros:

- High durability – 99.999999999% durability
- High availability – 99.99% availability over a given year
- High performance – low latency and high throughput
- Great option for frequently accessed data that needs high durability

Cons:

- Not the cheapest storage cost or request cost

# Amazon S3 Storage Classes – Standard – Infrequent Access (Standard_IA)

Standard – Infrequent Access storage is used for less frequently accessed data that still requires fast access when needed:

Pros:

- High durability – 99.999999999% durability
- Lower request costs
- Great option for data that needs to be readily available but not frequently accessed
- High performance like Standard storage

Cons:

- Higher request costs
- Lower availability – 99.9% availability over a given year

# Amazon S3 Storage Classes – Reduced Redundancy Storage (RRS)

Reduced Redundancy Storage (RRS) reduces costs by storing data in a less durable manner:

Pros:

- High availability – 99.99% availability over a given year
- Low retrieval costs

Cons:

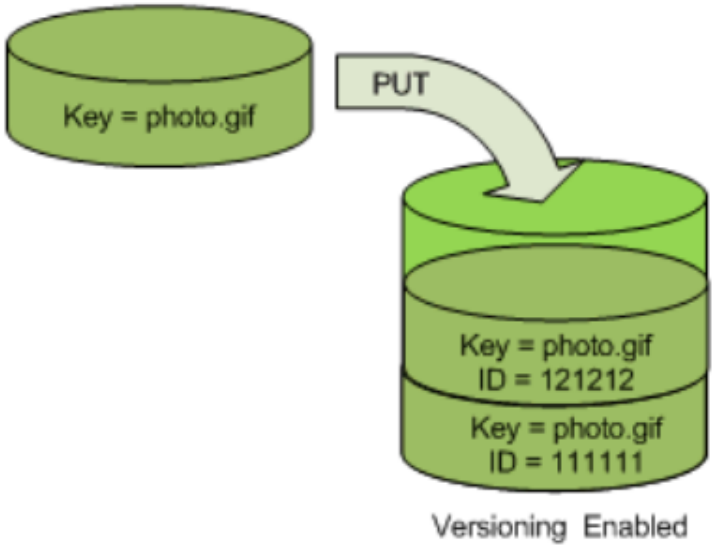
- Low durability – 99.99% durability
- Data *cannot* sustain 2 concurrent facility failures unlike the other two classes

# Amazon S3 – Versioning

- Allows for multiple versions of an object
- Protects against unintended overwrites and deletions
- Automatically archives objects
- Versioning is at the bucket level
- Configured via the Console or SDKs

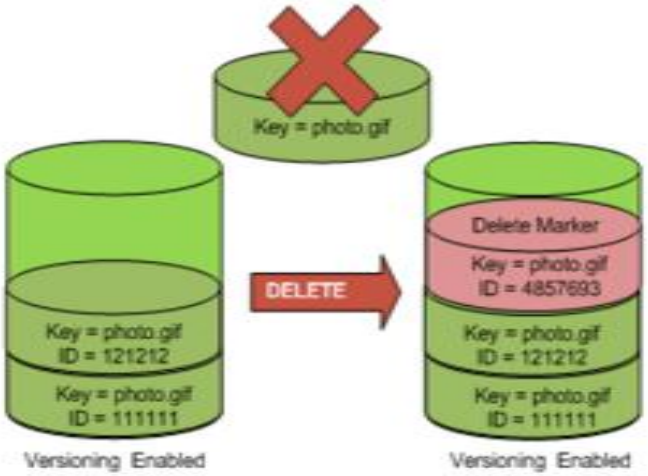
# Enabling Versioning

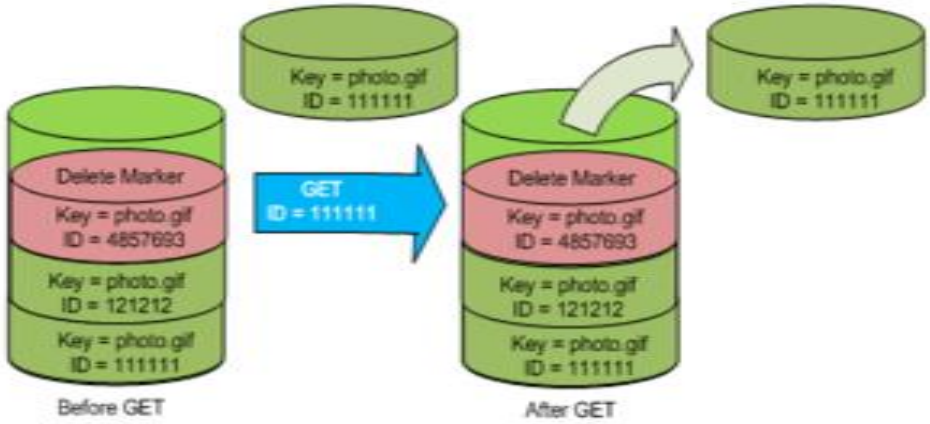
- Existing objects are unchanged
- Added objects are given unique version IDs
  - Automatically generated by S3



# Deleting versioned objects

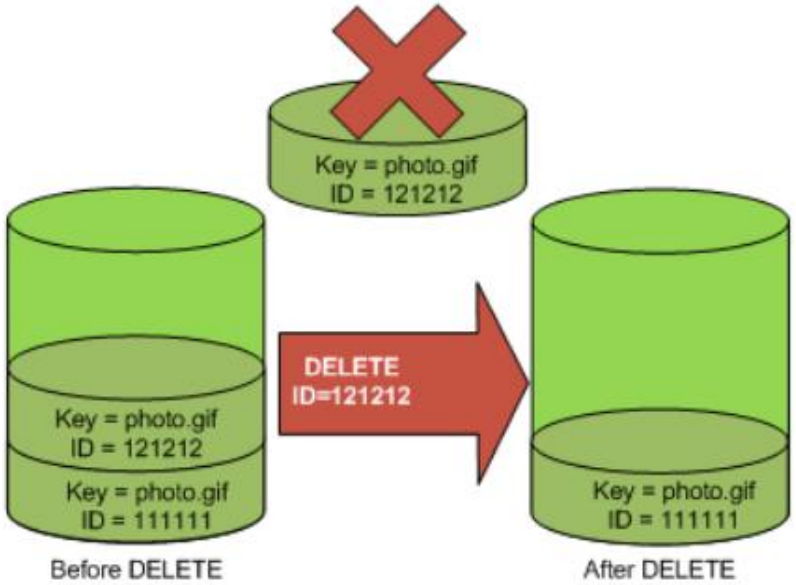
- All versions remain in the bucket but S3 inserts a delete marker
- The delete marker becomes the current version
- By default, **GET** requests retrieve the latest version
  - If the current version has a delete marker, it returns a **404 Not Found** error
  - You can get previous versions, however, by specifying an ID



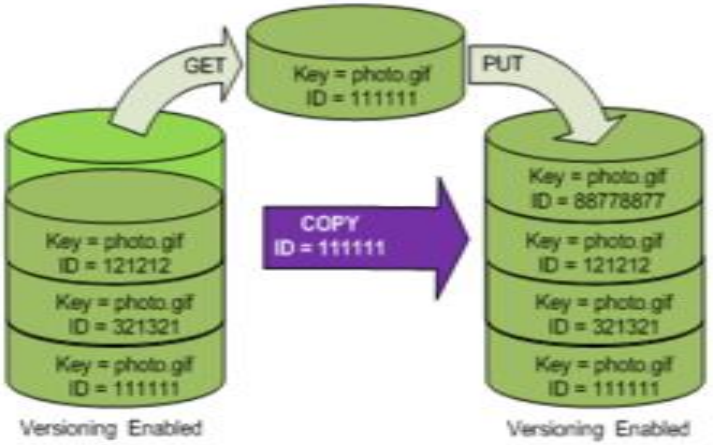
# Permanently Deleting

- To permanently delete a version, specify an ID



# Restoring versioned objects

- Any earlier version can be restored:
  - Copying a previous version into the same bucket will restore it as the current version
  - Permanently delete the current version (by specifying its ID)
- Copying an earlier version **GET**s the version and **PUT**s it in the bucket, giving it a new ID
  - That new ID is the used as the current version



# Using Lifecycle Management with Versioning

- Lifecycle Management can dictate what happens to versions after a certain amount of time
  - Example: Send noncurrent versions to Amazon Glacier after 30 days
  - Example: Permanently delete objects 180 days after they become noncurrent

- Can have separate policies for current and noncurrent versions

## Disabling Versioning

- Once version-enabled, a bucket *cannot* go back to being in an unversioned state

- You *can* suspend versioning
  - Future objects are given a version ID of null
  - Already versioned objects do not change

- To permanently delete a version, specify an ID

# Lifecycle Management

We can manage an object's lifecycle by using lifecycle configurations which tell Amazon S3 how to manage objects during their lifetime (ie: 30, 60, 90 days).

Lifecycle configurations are XML documents made up of a set of rules with actions we want Amazon S3 to perform on objects. These actions include:

- Transition actions where we define when objects transition to another S3 storage class
- Expiration actions where we can specify when an object expires and should be deleted

```
<LifecycleConfiguration>
  <Rule>
    <ID>example</ID>
    <Prefix>logs/</Prefix>
    <Status>Enabled</Status>
    <Transition>
      <Days>30</Days>
      <StorageClass>STANDARD_IA</StorageClass>
    </Transition>
    <Transition>
      <Days>90</Days>
      <StorageClass>GLACIER</StorageClass>
    </Transition>
  </Rule>
</LifecycleConfiguration>
```

## Object Lifecycle Scenario

You are storing documents, images, videos, or audio files and these objects are used to generate lower quality versions to save on bandwidth costs.

The high quality version is stored in the Standard class, but the lower quality version is stored in Reduced Redundancy Storage to save on request costs.

Because the higher quality objects will be infrequently accessed, after 30 days, you move the high quality objects into Standard_IA to save on storage costs.

# Lifecycle Management Limitations

Some of lifecycle management's limitations to be aware of:

- Objects must be stored at least 30 days in the current storage class before they can transition to Standard_IA
- You cannot transition from Standard_IA to Standard or Reduced Redundancy
- You cannot transition from Glacier to any other storage class
- You cannot transition from any storage class to Reduced Redundancy