AWS Foundation

SIMPLE STORAGE SERVICE (S3)

Agenda

1 Pre-S3 Scenarios

2 Introduction to S3

3 Storage Hierarchy

4 Buckets and Objects

5 Metadata and Storge Classes

6 Versioning

7 Lifecycle Management

8 Data Encryption and Logging

9 S3 CORS

10 S3 Object Lambda

Pre-S3 Scenarios

Online Cloud Storage

We can upload files, folders, images, songs, and videos from a machine and access them from anywhere in the world

Dropbox

iCloud

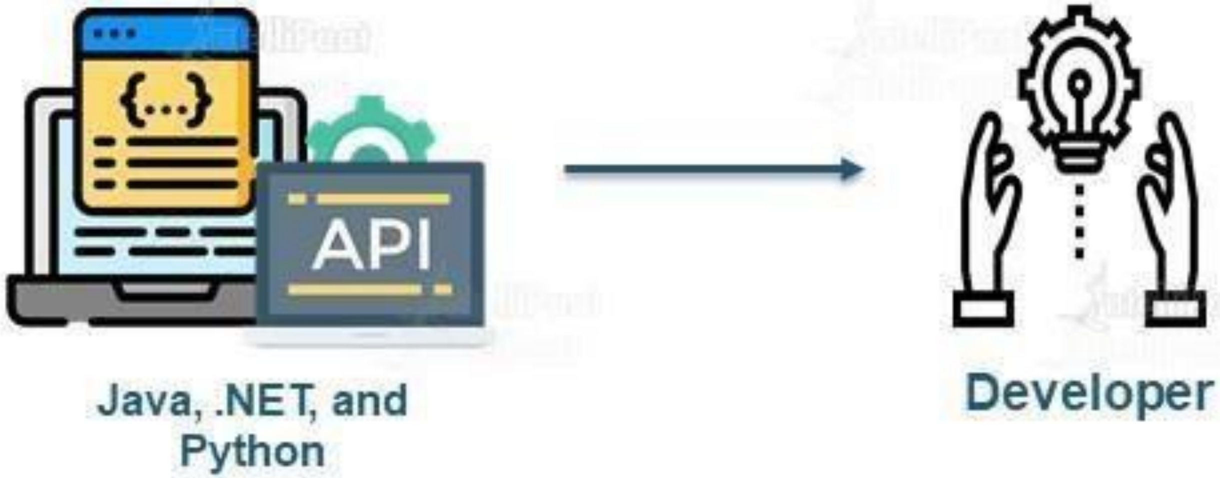
OneDrive

Google Drive

APIs

What is API?

- An **API** is a list of specifications that describe how information is exchanged between programs
- Software that wants to access another will call the API published by the other program

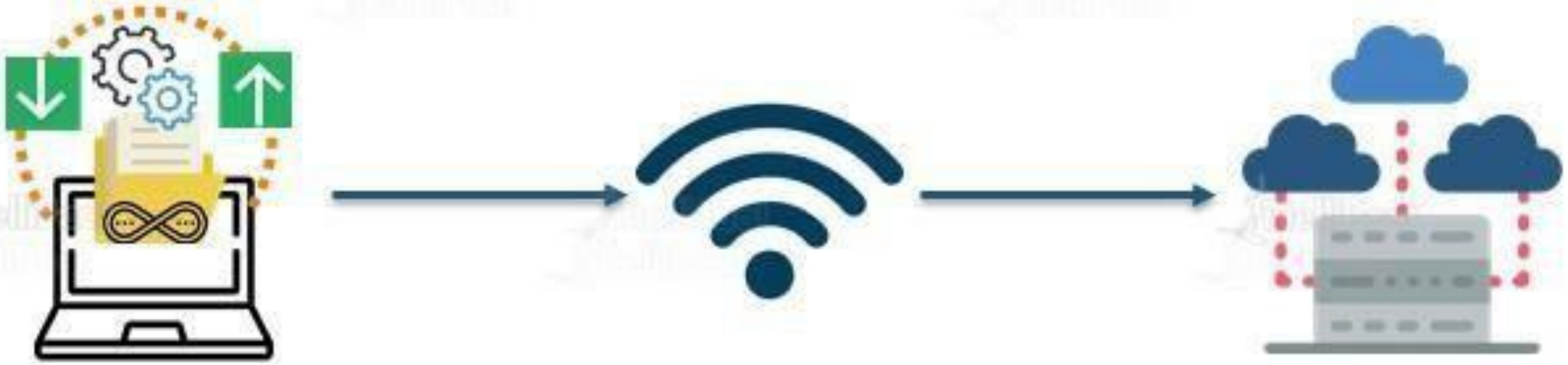


S3 Introduction

S3 Concepts

Simple Storage Service

- Amazon Simple Storage Service (S3) is a storage that can be maintained and accessed over the Internet
- S3 provides the web service that can be used to store and retrieve unlimited amount of data. Same can be done programmatically using Amazon-provided APIs



S3 Consistency Models

S3 Data Consistency Model

- S3 provides highly durable and available solutions by replicating all data in multiple data centers in a region
- Data uploaded in a particular region never leaves it
- Read-after-write consistency
- Eventual consistency

Replication of data in multiple data centers

Data uploaded never leaves the data center

Read after write

Eventual consistency

S3 Data Consistency Model

Consistent Read	Eventual Consistent Read
No stale reads	Stale reads are possible
Higher comparative read latency	Lower comparative read latency
Read throughput is comparatively lower	Read throughput is the highest

Replication of data in multiple data centers

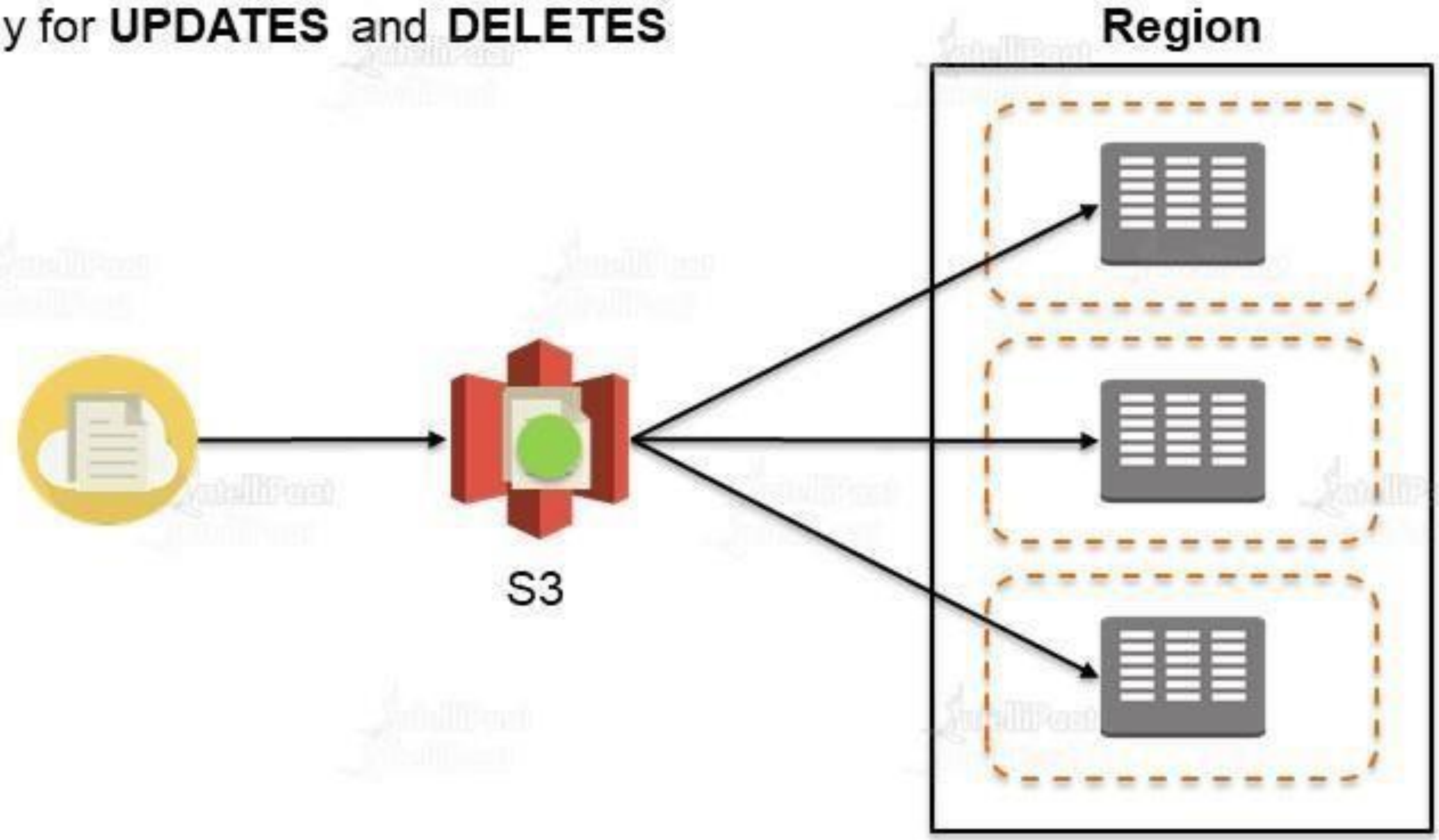
Data uploaded never leaves the data center

Read after write

Eventual consistency

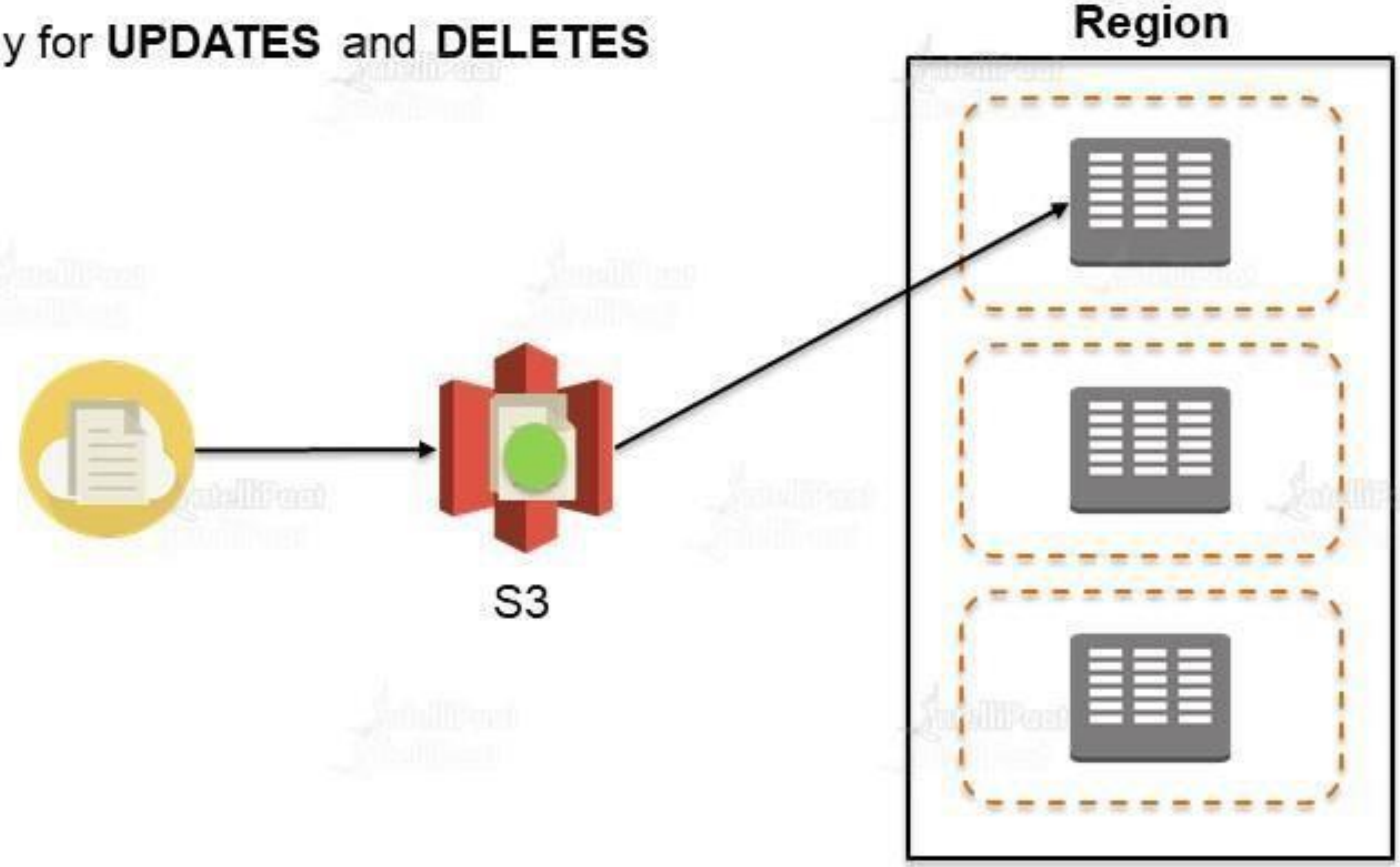
Consistency Model

Eventual consistency for **UPDATES** and **DELETES**



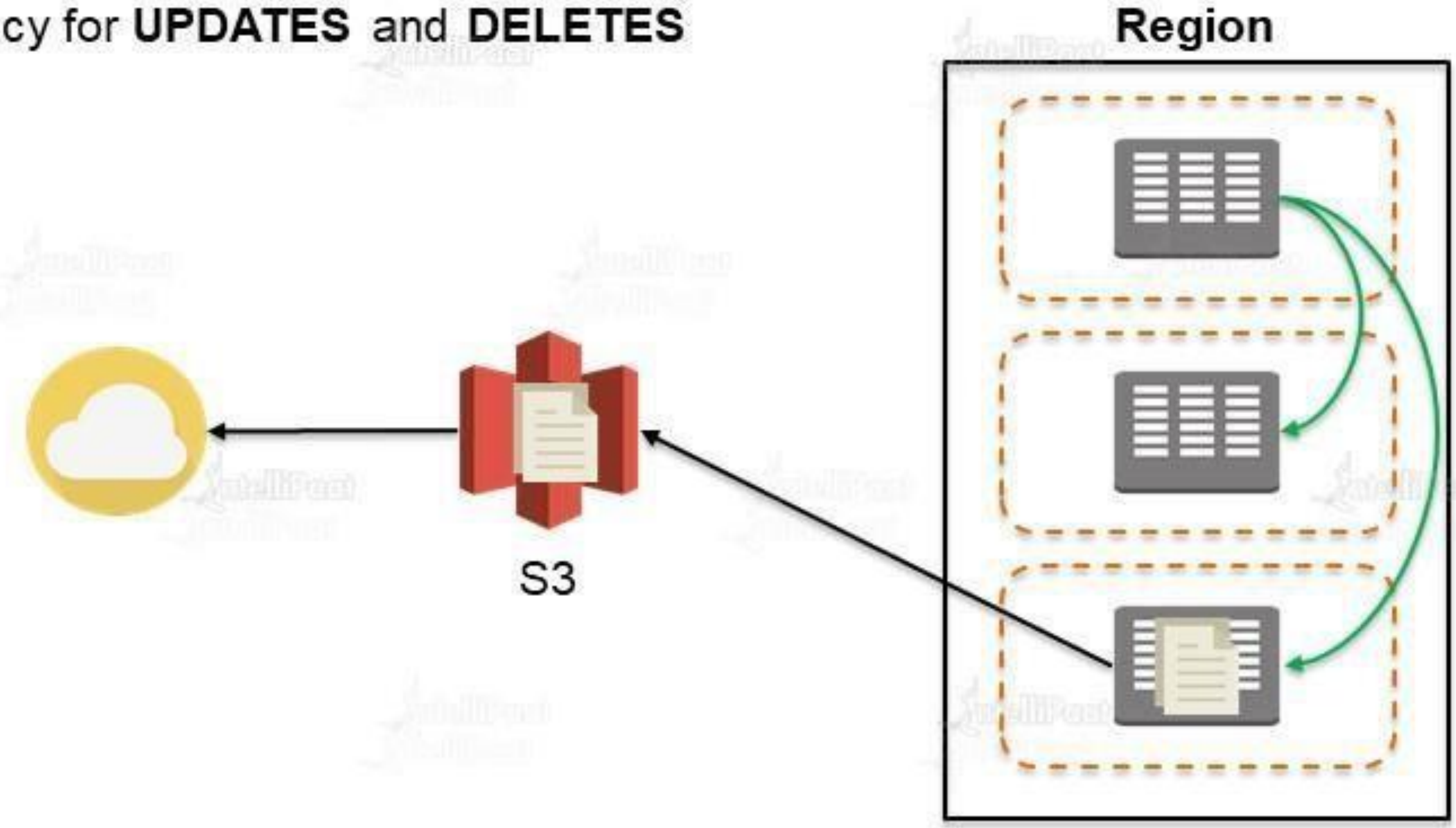
Consistency Model

Eventual consistency for **UPDATES** and **DELETES**



Consistency Model

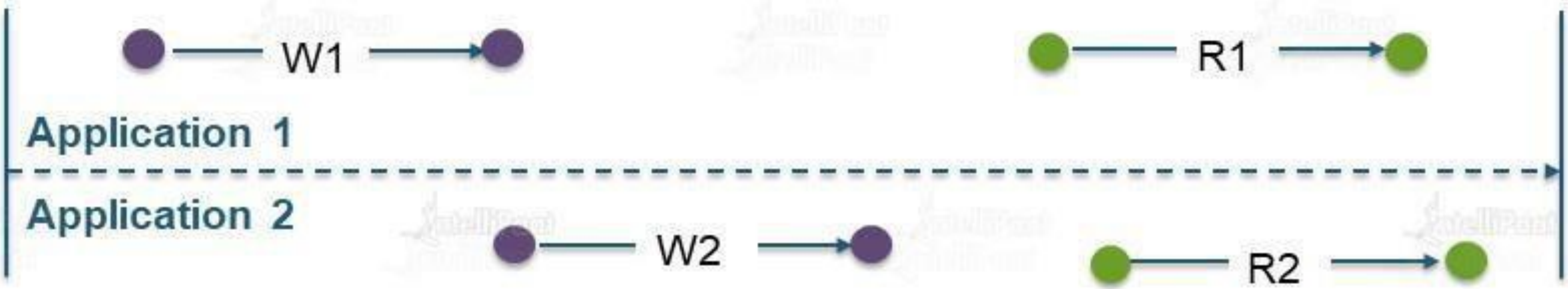
Eventual consistency for **UPDATES** and **DELETES**



S3 Consistency Examples

Example 1

- Concurrent applications

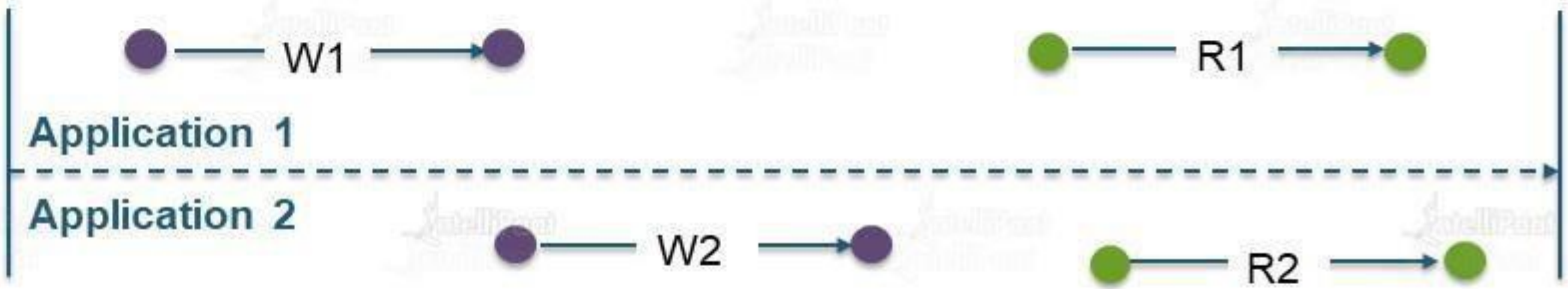


- W1 → Name: "EC2", W2 → Name: "EBS"
- R1 → consistent – Name: "EBS", eventual – Name: "EBS" or "EC2" or Nothing
- R2 → consistent – Name: "EBS", eventual – Name: "EBS" or "EC2" or Nothing

S3 Consistency Examples

Example 2

- Concurrent applications

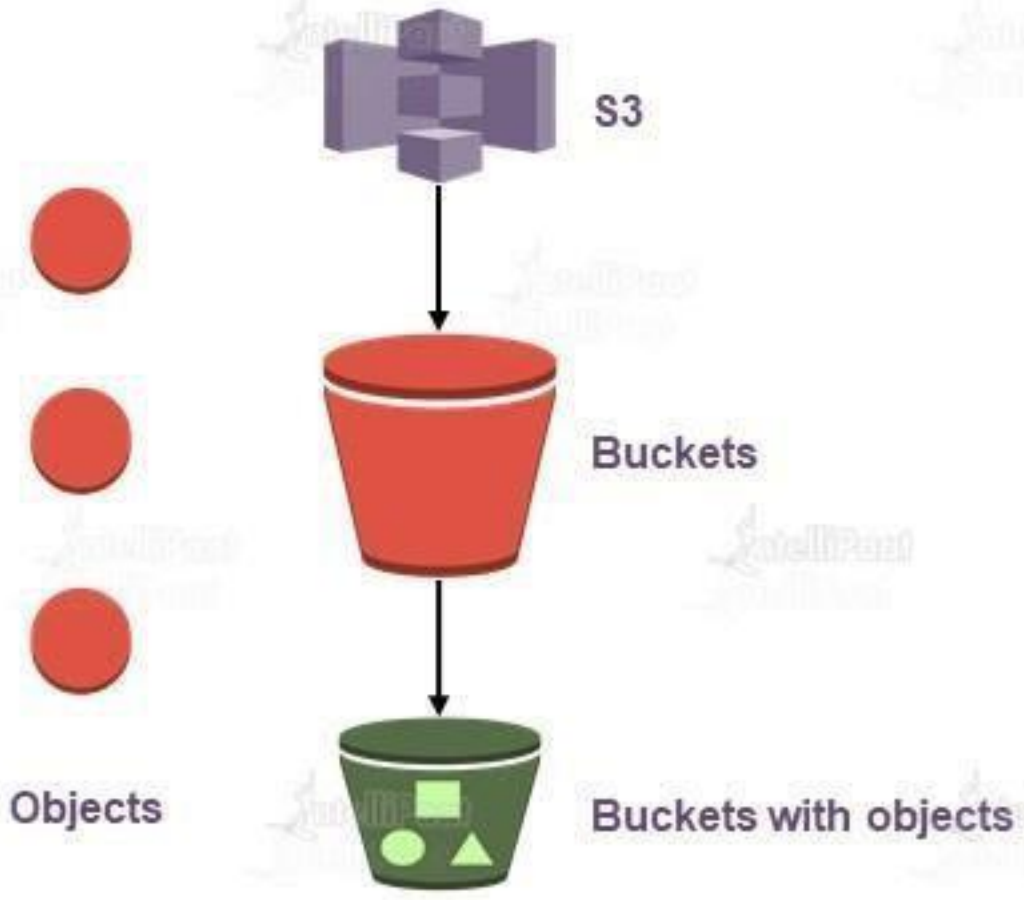


- W1 → Name: “EC2”, W2 → Name: “EBS”
- R1 → consistent – Name: “EBS” or “EC2”, eventual – Name: “EBS” or “EC2” or Nothing
- R2 → consistent – Name: “EBS”, eventual – Name: “EBS” or “EC2” or Nothing

Storage Hierarchy

Storage Hierarchy

- S3 follows a storage hierarchy while keeping data (documents, images, videos, files, etc.)
- Management Console or S3 APIs can be used to manage buckets and objects



Buckets

Buckets

Bucket count & restrictions

Communicating using SDK

Accessing buckets

Naming convention

By default, the maximum number of buckets that can be created per account is 100. For additional buckets, one can submit a service limit increase

Buckets

- Bucket count & restrictions
- **Communicating using SDK**
- Accessing buckets
- Naming convention

While using AWS SDKs, first a client is created, and then this client is used to send request to create a bucket. The client is created by specifying an AWS region, and the client uses an endpoint to communicate with Amazon S3

For Example:

If a client is created by specifying the N. Virginia (default) region, then the following endpoint is used to communicate with Amazon S3:

```
s3.amazonaws.com
```

For any other region:

```
– s3<region>.amazonaws.com
```

Buckets

Bucket count & restrictions

Communicating using SDK

Accessing buckets

Naming convention

Types of URLs to Access Buckets

- Virtual hosted style:
  http://bucket.s3.amazonaws.com/object OR
  http://bucket.s3-aws-region.amazonaws.com/object
- Path style:
  http://s3.amazonaws.com/bucket/object OR
  http://s3-aws-region.amazonaws.com/bucket/object

Buckets

Bucket count & restrictions

Communicating using SDK

Accessing buckets

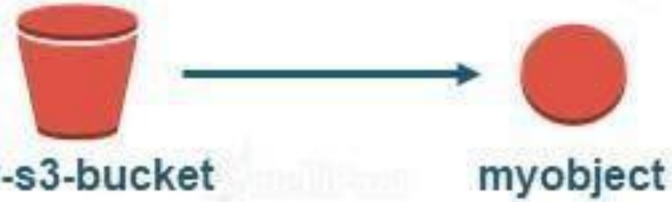
Naming convention

Bucket names have to be globally unique irrespective of the region they are created in. As buckets can be accessed using URLs, it is recommended that bucket names follow DNS naming conventions, i.e., all letters should be in lowercase

Objects

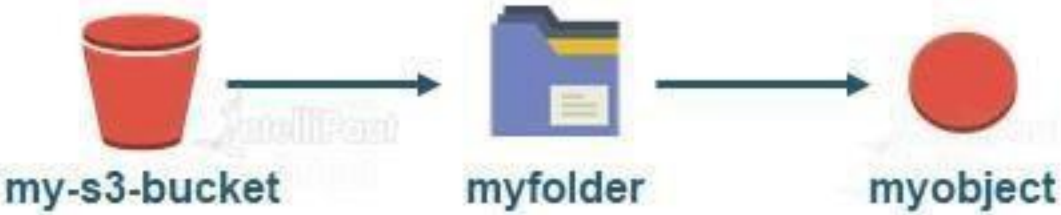
Objects

- When there is no folder, and an object resides in the bucket:



http://my-s3-bucket.s3.amazonaws.com/**myobject**

- When there is a folder on console, and the folder name is used as prefix with the object key:



http://my-s3-bucket.s3.amazonaws.com/myfolder/**myobject**

- Objects are videos, images, documents, etc., which are stored inside buckets
- While creating a bucket, a name is given, and the "name" is the object key
- There cannot be any sub-bucket or sub-folder inside a bucket (physically, however, folders can be created on the console, which provides a logical hierarchy only and are used as prefixes in the object key)

Metadata and Storage Class

Metadata and Storage Classes

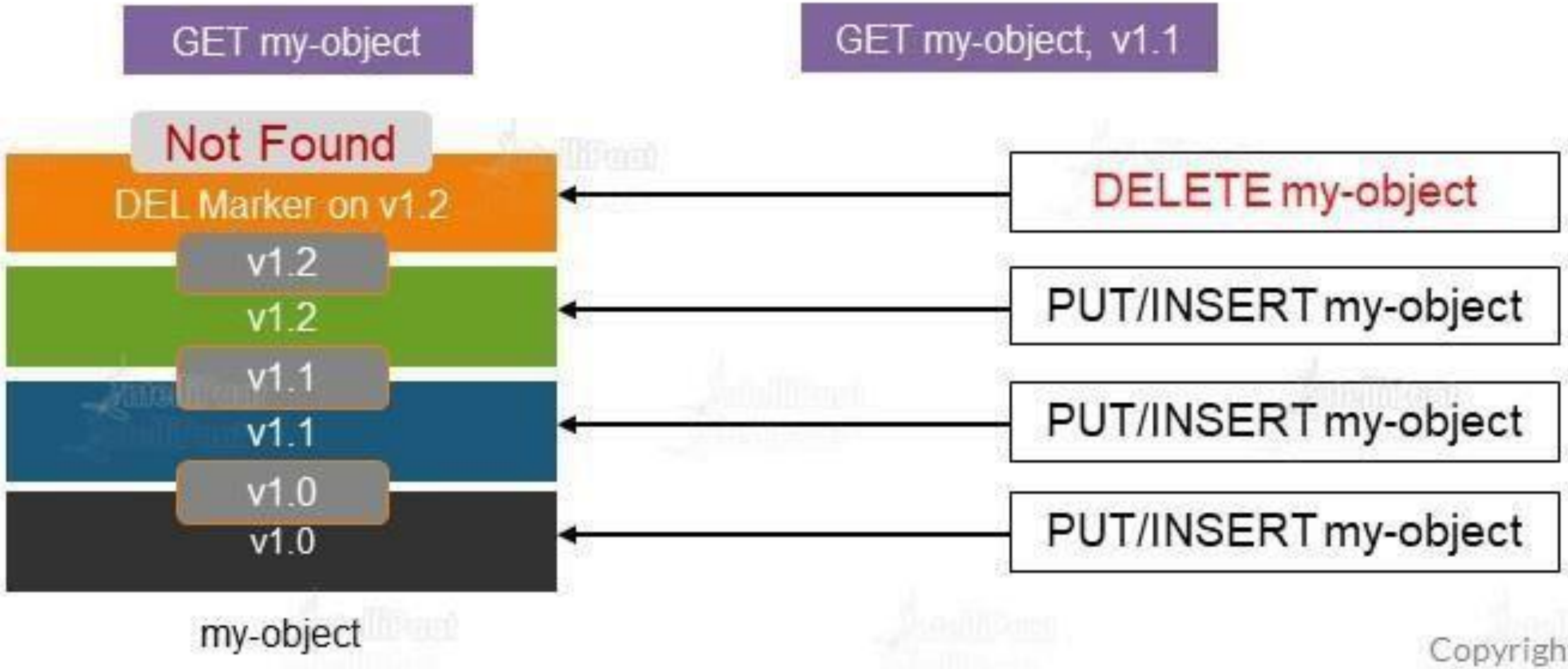
- Object metadata: For each object, S3 maintains a set of system metadata
  - Date: Current date and time
  - Content-length: Object size in bytes
  - Last-modified: Object creation or last modified date
  - x-amz-server-side-encryption: Whether encryption is enabled or not
  - x-amz-version-id: Object version
  - x-amz-delete-marker: Whether the object is a delete marker in the case of versioning
  - x-storage-class: The storage class associated with the object

- Storage class: Each object has a storage class associated with it
  - STANDARD: For frequently accessed data. 11 9s of durability and 4 9s of availability
  - STANDARD IA: For less frequently accessed real-time data. 11 9s of durability and 3 9s of availability
  - REDUCED REDUNDANCY: For non-critical, reproducible data with lower levels of redundancy than the standard storage class. 4 9s of durability and 4 9s of availability

Versioning

Versioning

- Versioning enables us to keep multiple versions of the same object in one bucket
- Versioning has to be enabled explicitly. Each object has a version ID
- Existing objects are not overwritten



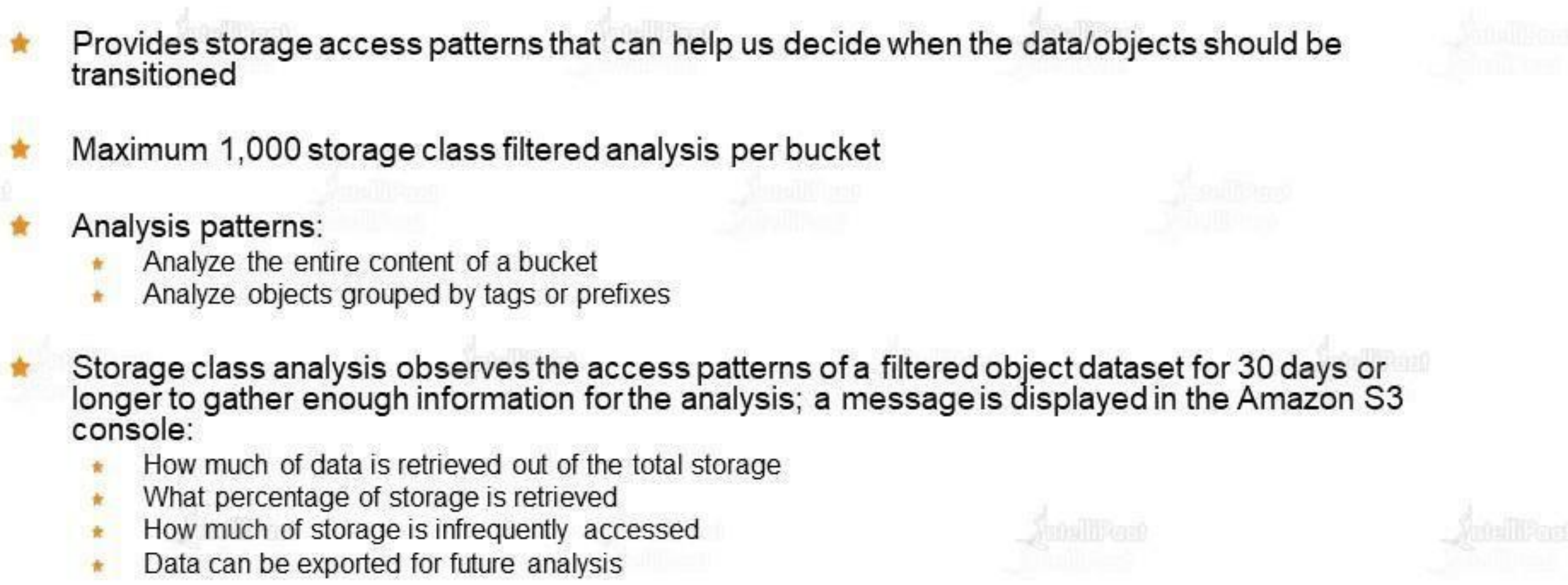
Lifecycle Management

Lifecycle Management

- Lifecycle Management works at the bucket level, enabling us to perform an action on objects based on rules
- Actions
  - Transition: Objects are transitioned from one storage class to another
    - STANDARD or REDUCED REDUNDANCY to STANDARD_IA
    - STANDARD to GLACIER
    - Objects must be stored for at least 30 days in the current storage class before transitioning
  - Expiration: Objects are expired and deleted

Storage Class Analysis

Storage Class Analysis



- Provides storage access patterns that can help us decide when the data/objects should be transitioned
- Maximum 1,000 storage class filtered analysis per bucket
- Analysis patterns:
  - Analyze the entire content of a bucket
  - Analyze objects grouped by tags or prefixes
- Storage class analysis observes the access patterns of a filtered object dataset for 30 days or longer to gather enough information for the analysis; a message is displayed in the Amazon S3 console:
  - How much of data is retrieved out of the total storage
  - What percentage of storage is retrieved
  - How much of storage is infrequently accessed
  - Data can be exported for future analysis

Storage Class Analysis

STANDARD/ REDUCED_REDUNDANCY

ONEZONE_IA

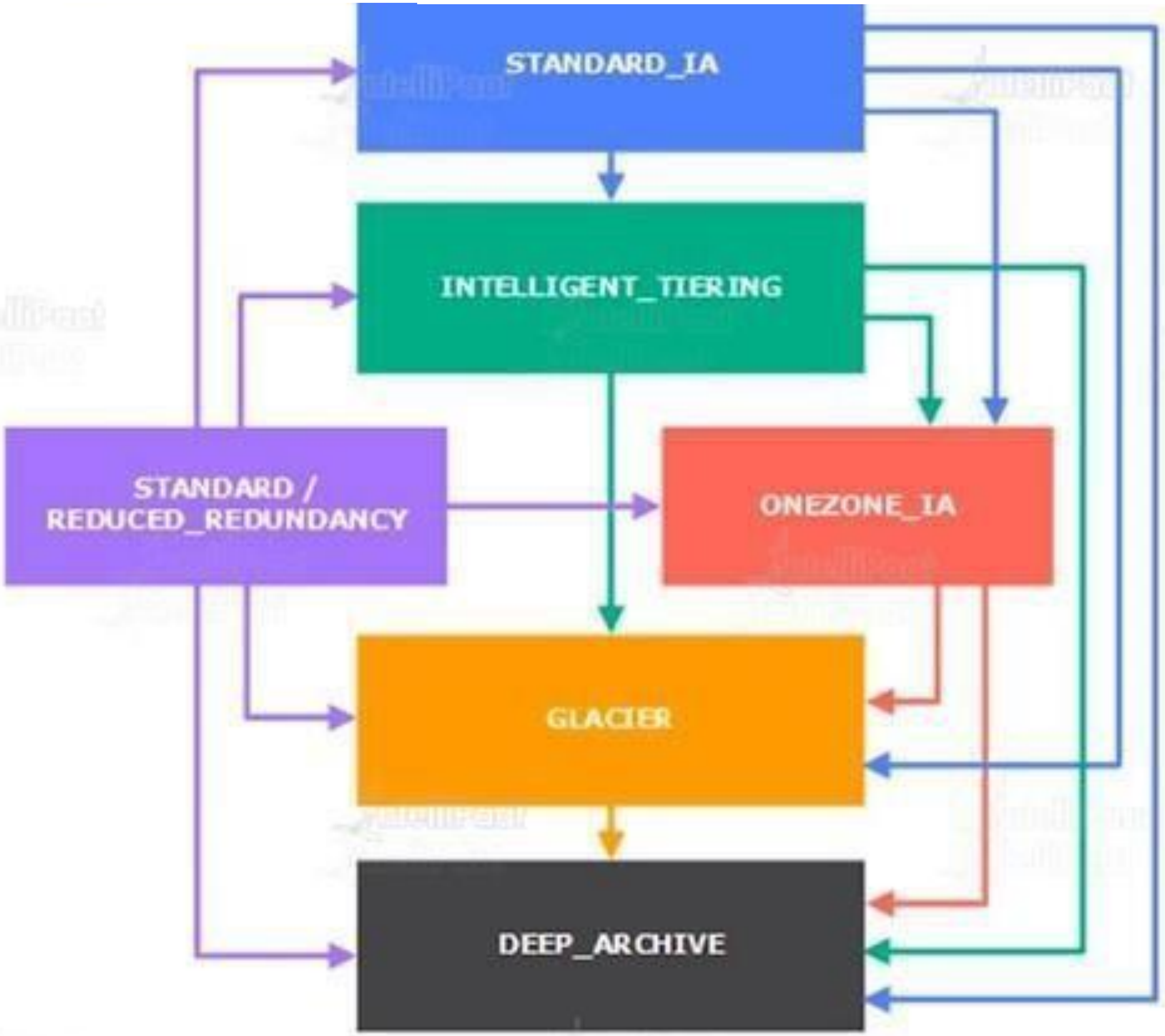
GLACIER

STANDARD_IA

INTELLIGENT_TIERING

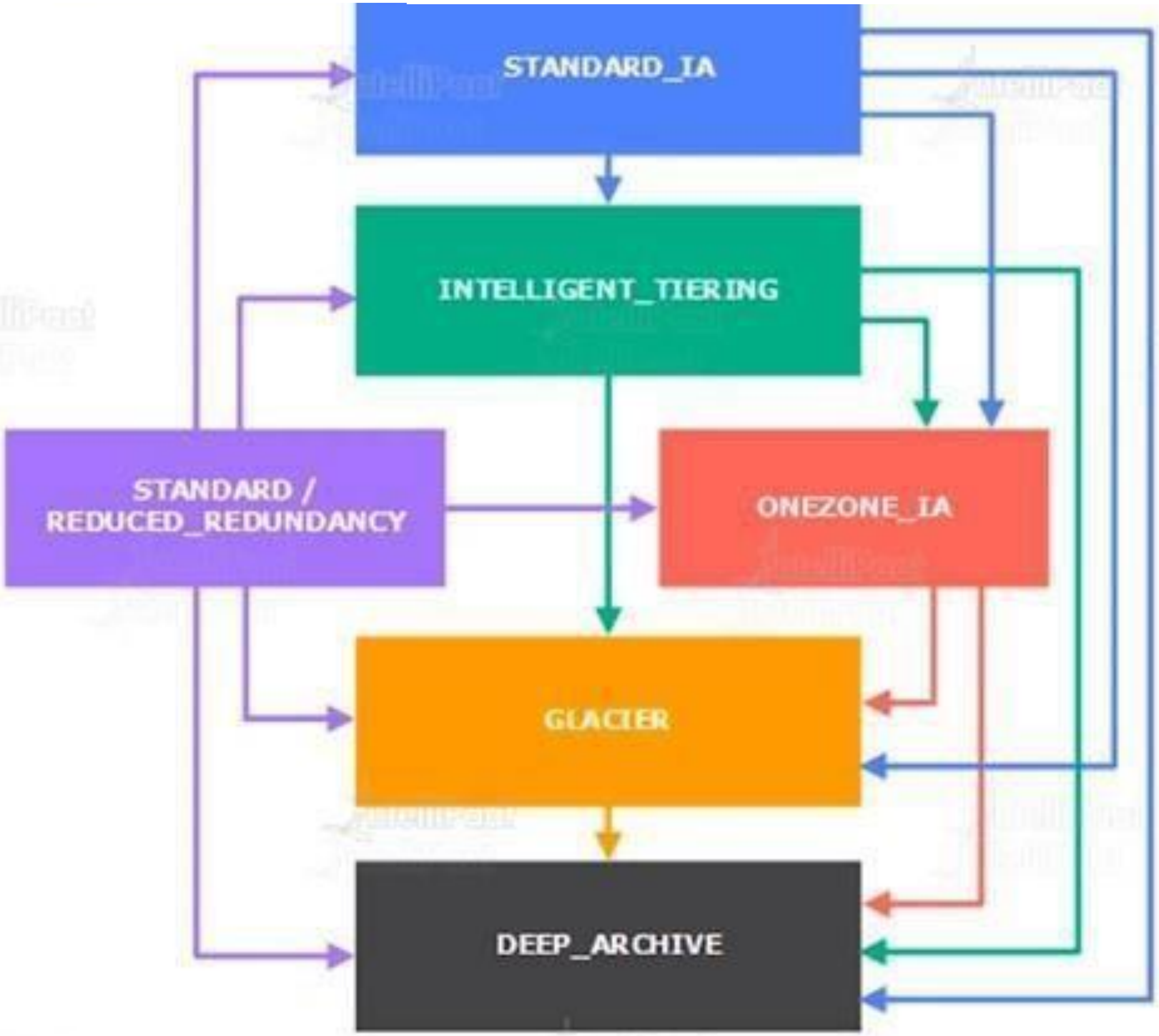
DEEP_ARCHIVE

Storage Class Analysis



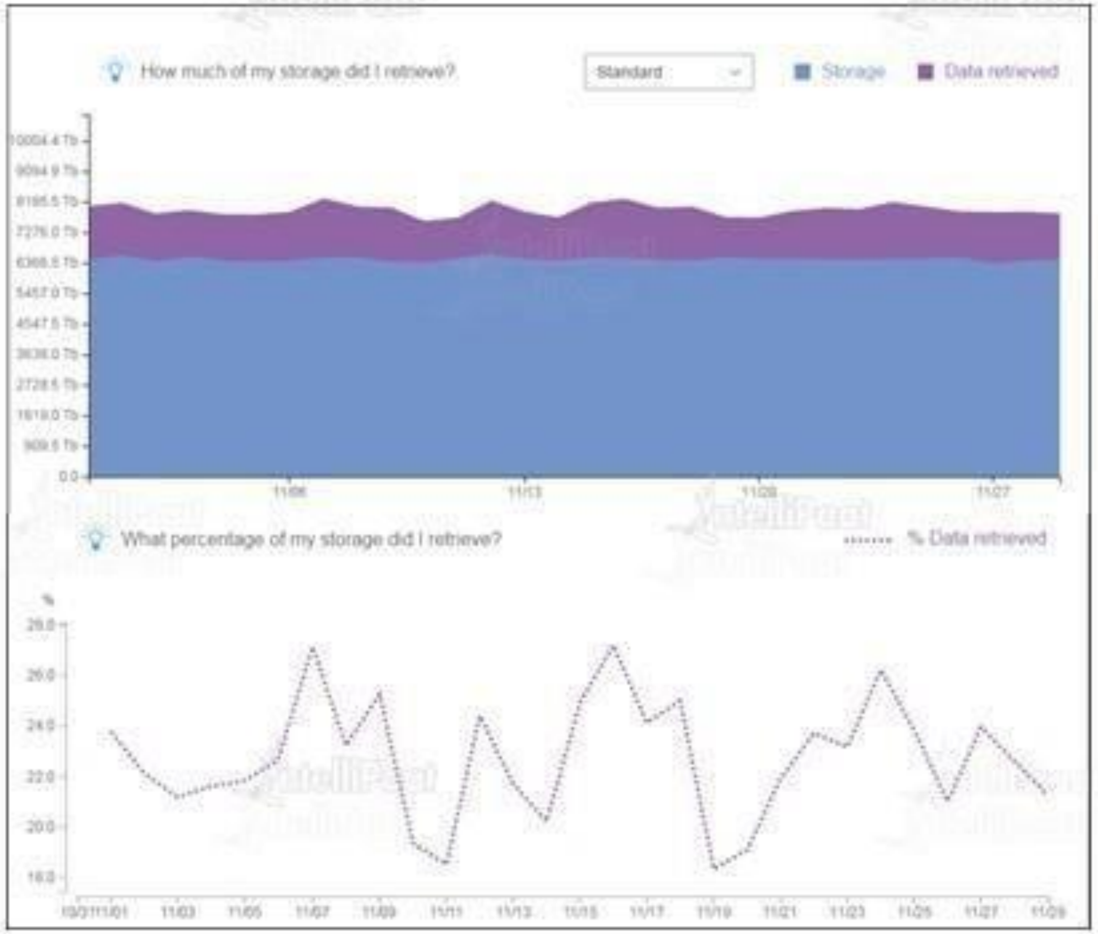
- ✓ STANDARD storage class to any other storage class
- ✓ Any storage class to the GLACIER or DEEP_ARCHIVE storage classes
- ✓ GLACIER storage class to the DEEP_ARCHIVE storage class

Storage Class Analysis



- ☒ Any storage class to the STANDARD storage class
- ☒ INTELLIGENT_TIERING storage class to the STANDARD_IA storage class
- ☒ DEEP_ARCHIVE storage class to any other storage class

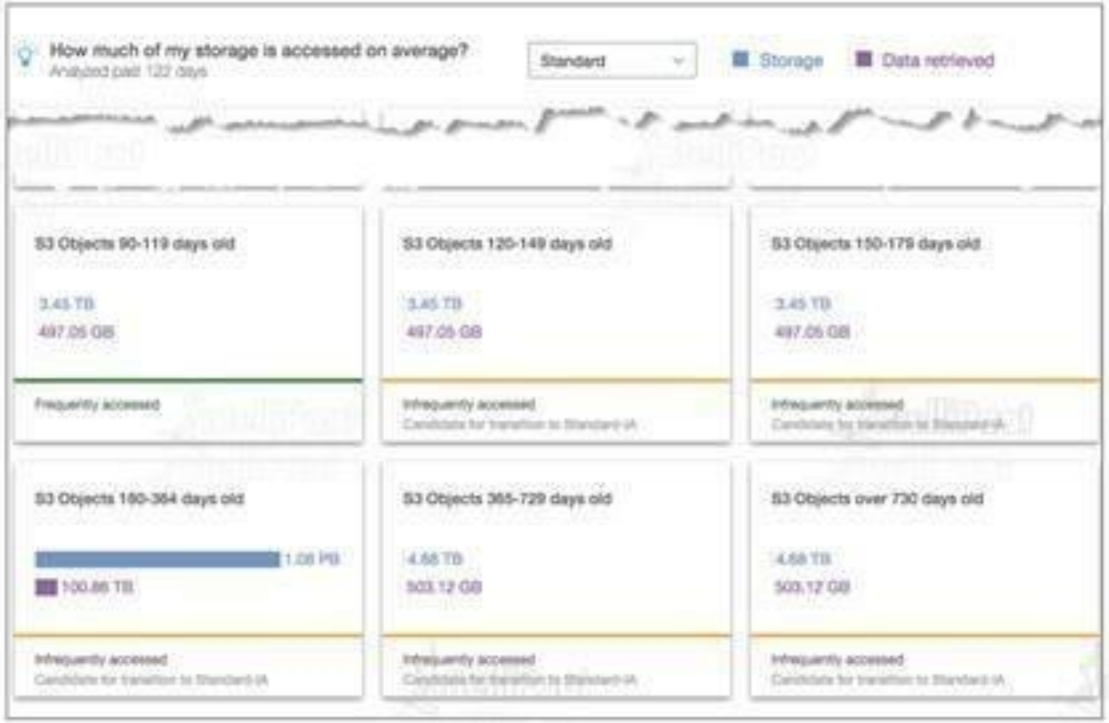
Storage Class Analysis



How much of my storage did I retrieve?

Standard

What percentage of my storage did I retrieve?



How much of my storage is accessed on average?
Analyzed past 122 days

Standard

S3 Objects 90-119 days old

3.45 TB
497.05 GB

Frequently accessed

S3 Objects 120-149 days old

3.45 TB
497.05 GB

Infrequently accessed
Candidate for transition to Standard-IA

S3 Objects 150-179 days old

3.45 TB
497.05 GB

Infrequently accessed
Candidate for transition to Standard-IA

S3 Objects 180-364 days old

1.08 PB
100.86 TB

Infrequently accessed
Candidate for transition to Standard-IA

S3 Objects 365-729 days old

4.68 TB
503.12 GB

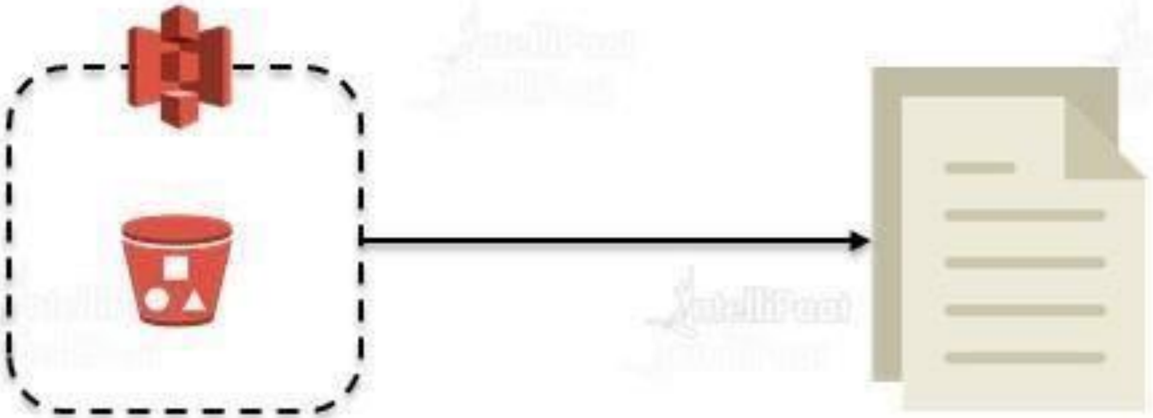
Infrequently accessed
Candidate for transition to Standard-IA

S3 Objects over 730 days old

4.68 TB
503.12 GB

Infrequently accessed
Candidate for transition to Standard-IA

Inventory



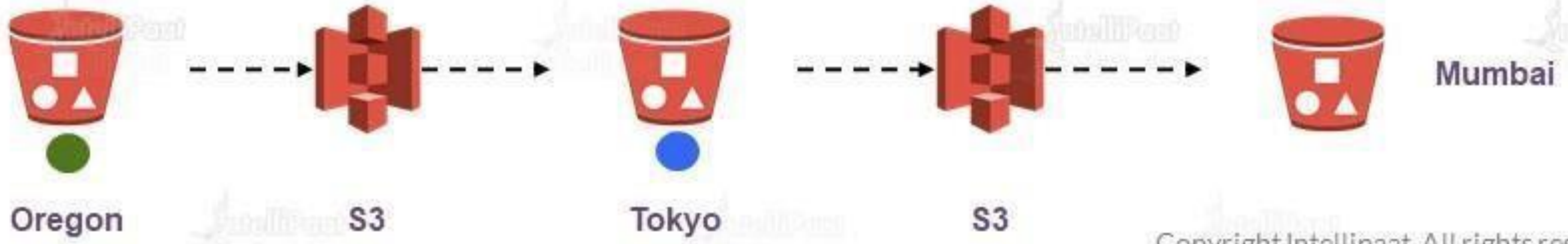
- Inventory provides a report and its metadata for objects on a daily or weekly basis in a comma-separated output file
- Metadata output is configurable
- Source bucket: For which the inventory is created
- Destination bucket: Wherein the inventory is stored

Cross-Region replication

Cross-Region replication

Automatic asynchronous replication of objects to a different region

- The subset of objects can also be replicated using prefix matches
- Versioning should be enabled for CRR to work
- The source bucket or its objects can be replicated to only one target bucket
- The deletion of a specific object version is not replicated over to the other region
- The existing objects of a bucket are not replicated (if replication is enabled later on)
- Lifecycle Management actions are not replicated
- Replicated objects are not replicated to other regions



Cross-Region replication

- Compliance requirements
- Latency
- Operational
- Ownership

Data Encryption

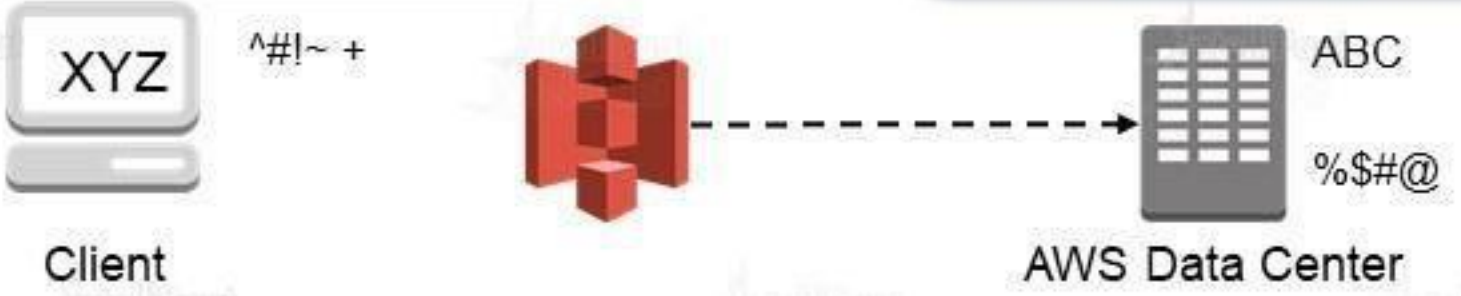
Data Encryption

Server-side encryption

- S3 encrypts data at the object level as it writes to disks in its data centers and decrypts it when accessed. "x-amz-server-side-encryption-"
- SSE-S3 □ x-amz-server-side-encryption:AES-256
- SSE-KMS □ x-amz-server-side-encryption-aws-kms-key-id:<kms_key_id>
- SSE-C □ customer algorithm, customer key, and customer key MD are passed

Client-side encryption

- Client-side encryption refers to encrypting data before sending it to Amazon S3. Following two options are available for using data encryption keys:
  - AWS KMS-managed customer master key
  - Client-side master key



Server Access Logging

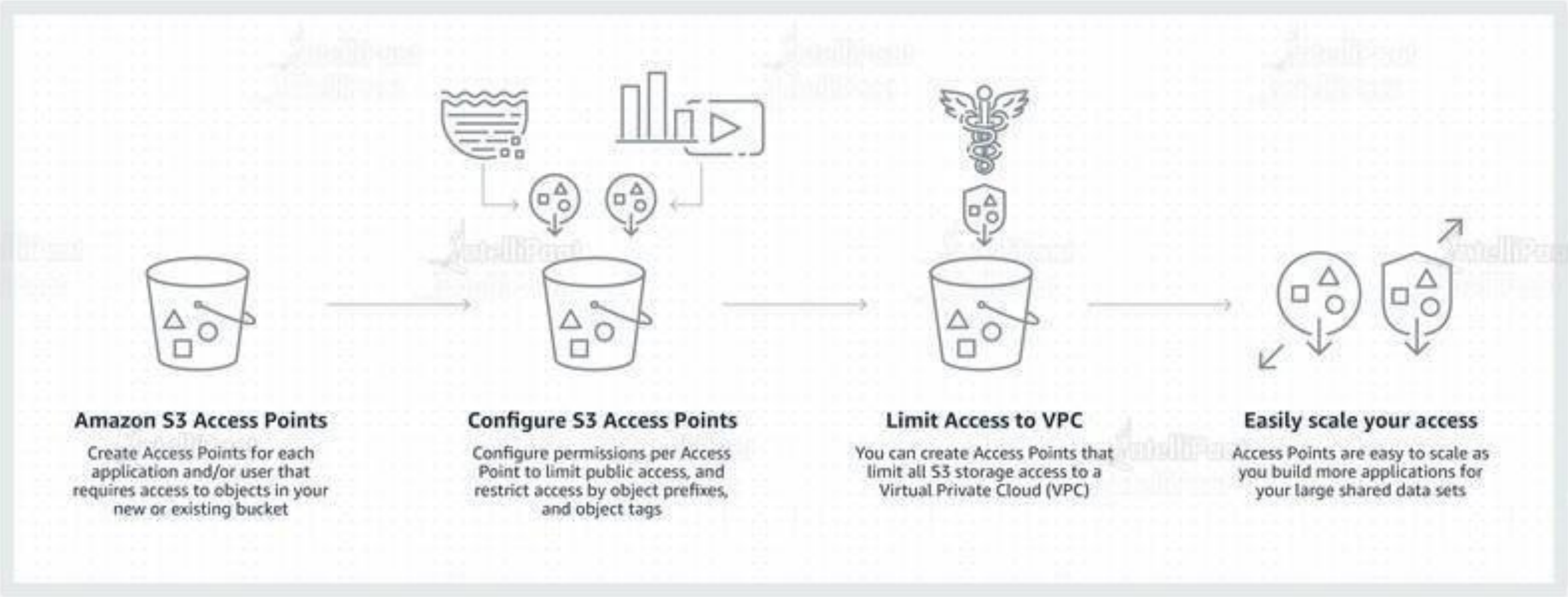
Server Access Logging

- Access logging enables us to track requests at the bucket level. Access logs are stored in separate buckets
- Access log format:
  - Bucket owner: The owner of the source bucket
  - Bucket name: The name of the bucket that the request was processed against
  - Time: The time at which the request was received
  - Remote IP: The IP address of the requestor
  - Requester: The ID of the requestor
  - Operation: REST.*http_method.resource_type*
  - Key: The object key in URL
  - Request-URI: The Request-URI part of the HTTP request message
  - HTTP status, error code, and bytes sent
  - Object size: The total size of the object in bytes
  - Total time: Measured in ms, from the time the request is received to the time the last byte of the response is sent
  - Turn-around time: The number of milliseconds that S3 spent, processing the request
  - Referrer, user agent, and the version ID

S3 Access Points

S3 Access Points

Access points are unique hostnames that customers create to enforce distinct permissions and network controls for any request made through the access point



Amazon S3 Access Points

Create Access Points for each application and/or user that requires access to objects in your new or existing bucket

Configure S3 Access Points

Configure permissions per Access Point to limit public access, and restrict access by object prefixes, and object tags

Limit Access to VPC

You can create Access Points that limit all S3 storage access to a Virtual Private Cloud (VPC)

Easily scale your access

Access Points are easy to scale as you build more applications for your large shared data sets

S3 Access Points

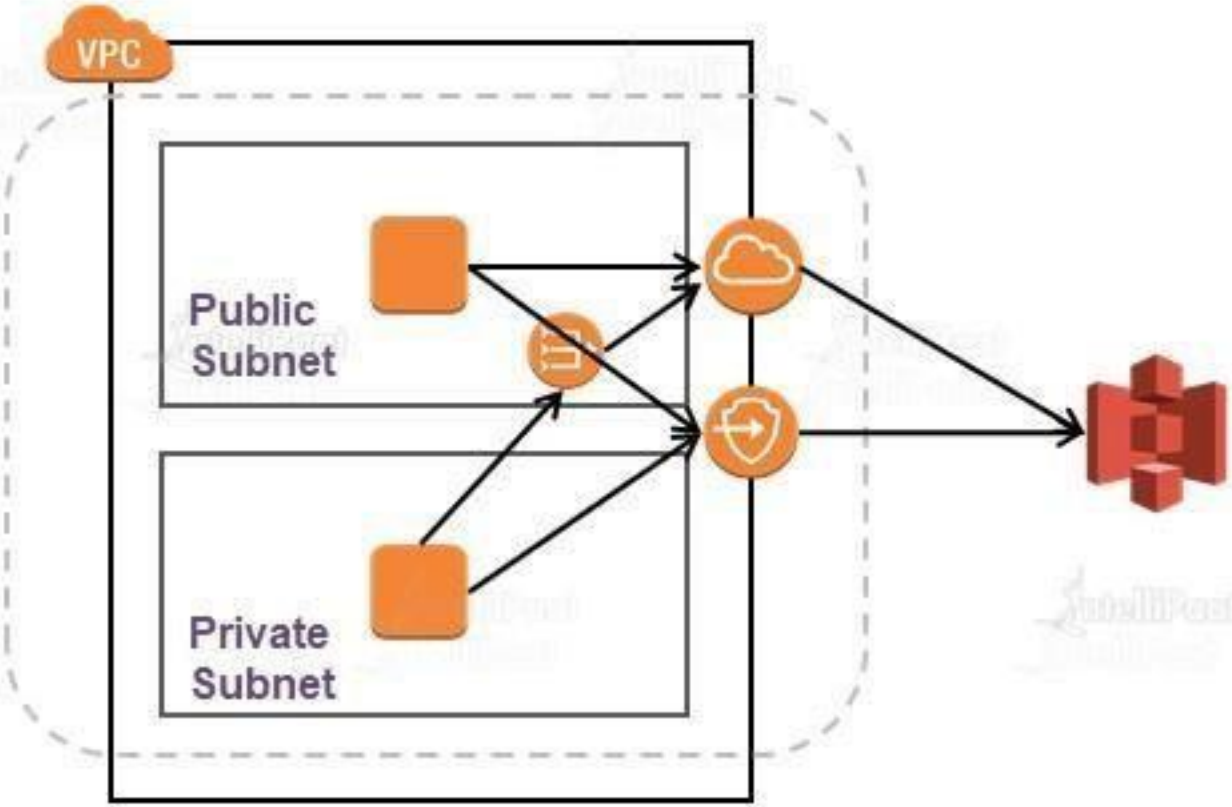
When should we consider using S3 access points

- Large shared datasets
- Restrict access to VPC
- Test new access policies
- Limit access to specific account IDs
- Provide a unique name

Connection Using VPC Endpoints

Connection using VPC Endpoints

Connect to S3 from EC2 instances in private subnets so that traffic never leaves Amazon's N/W



Connection using VPC Endpoints

- https://aws.amazon.com/s3/pricing/
- Storage
- Standard
  - US$0.023/GB for the first 50 TB/month
  - US$0.022/GB for the next 450 TB/month
  - US$0.021/GB for the next 500 TB/month
- Standard: IA: US$0.0125 per GB
- Glacier: US$0.004 per GB

- Requests
- PUT, COPY, POST, LIST: US$0.005 per 1000 requests
- GET and all others: US$0.0004 per 1000 requests

Total Storage: 750 TB

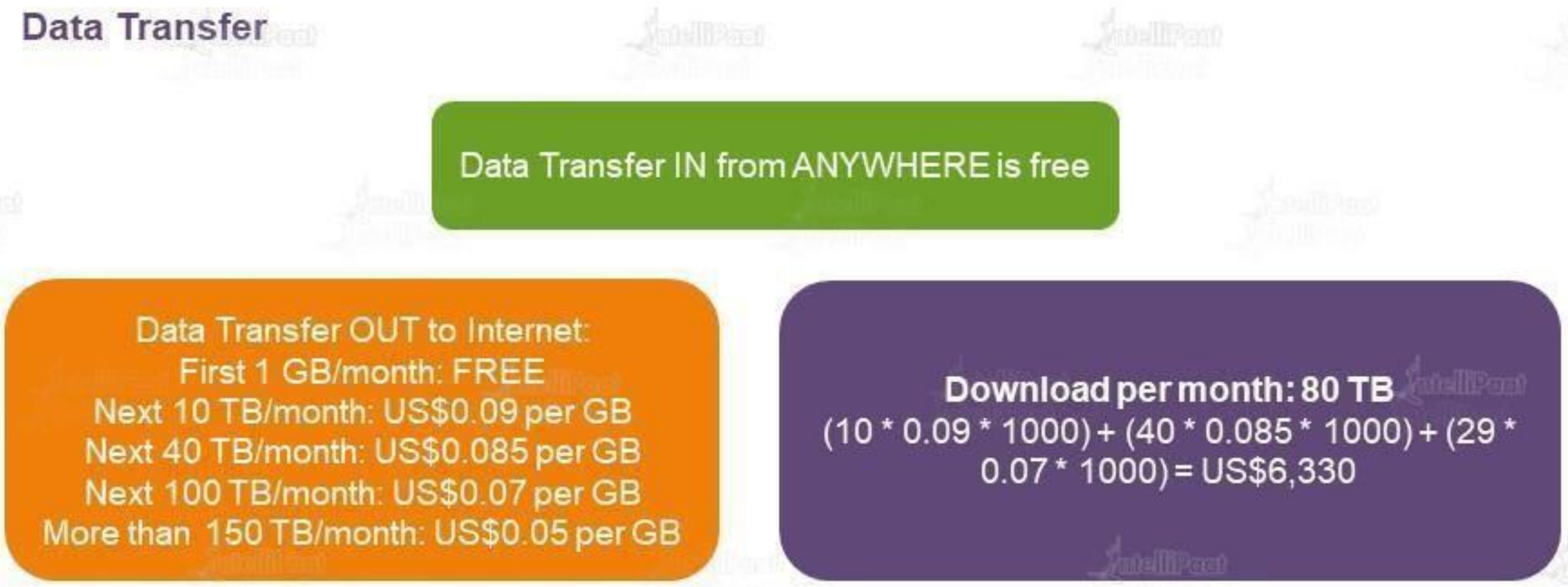
(50*0.023*1000) + (450*0.022*1000) + (250*0.021*1000) = US$16,300

150 million GET requests = (150,000,000/1000) * $0.0004 = US$6

500,000 PUT requests = (500000/1000) * $0.005 = US$2.5

S3 Pricing

S3 Pricing – us-east1



Data Transfer

Data Transfer IN from ANYWHERE is free

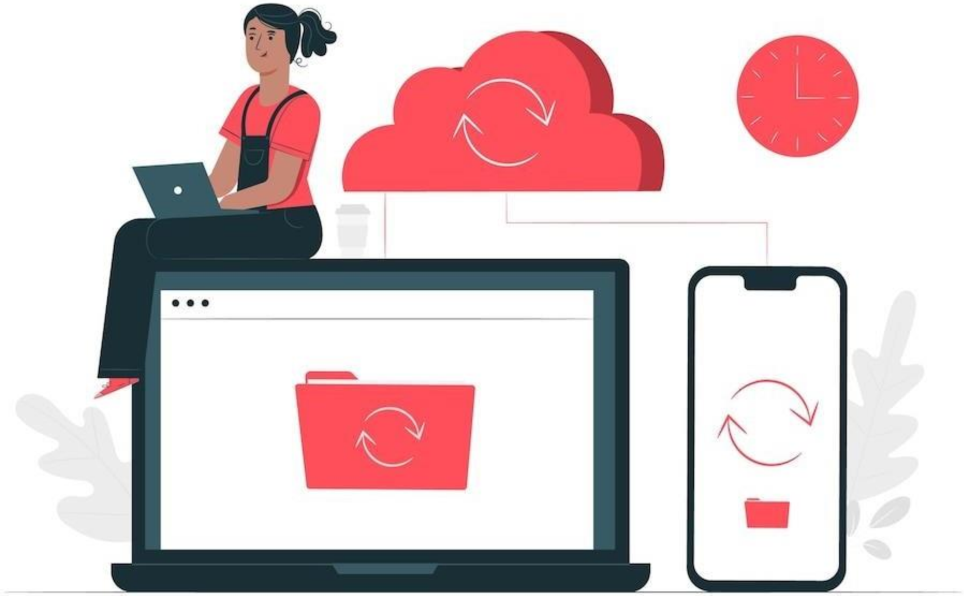
Data Transfer OUT to Internet:
First 1 GB/month: FREE
Next 10 TB/month: US$0.09 per GB
Next 40 TB/month: US$0.085 per GB
Next 100 TB/month: US$0.07 per GB
More than 150 TB/month: US$0.05 per GB

Download per month: 80 TB
(10 * 0.09 * 1000) + (40 * 0.085 * 1000) + (29 * 0.07 * 1000) = US$6,330

AWS CORS

AWS CORS

Cross-origin resource sharing (CORS) specifies how client web applications loaded in one domain can interact with resources in another domain. With CORS support, you can use Amazon S3 to create rich client-side web applications and selectively allow cross-origin access to your Amazon S3 resources.



AWS CORS

How does Amazon S3 evaluate S3 cors ?

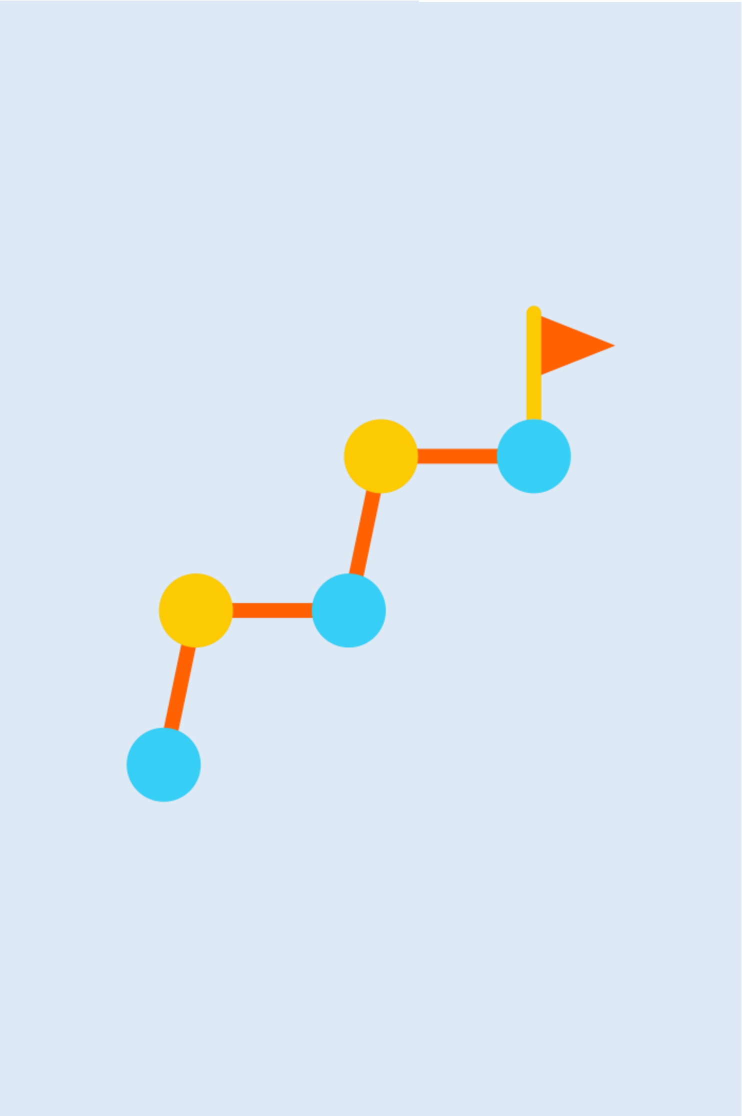
When Amazon S3 receives a browser preflight request, it evaluates the bucket's CORS configuration and uses the first CORS rule rule that matches the incoming browser request to enable a cross-origin request.The Origin header of the request must match an AllowedOrigin element.In the case of a preflight OPTIONS request, the request method (for example, GET or PUT) or the Access-Control-Request-Method header must be one of the AllowedMethod elements.

S3 Object Lambda

S3 Object Lambda

S3 Object Lambda is a new feature that allows you to add your own code to process data from S3 before returning it to an application. S3 Object Lambda integrates with your existing applications and uses AWS Lambda functions to process and transform your data as it is retrieved from S3.

S3 Object Lambda



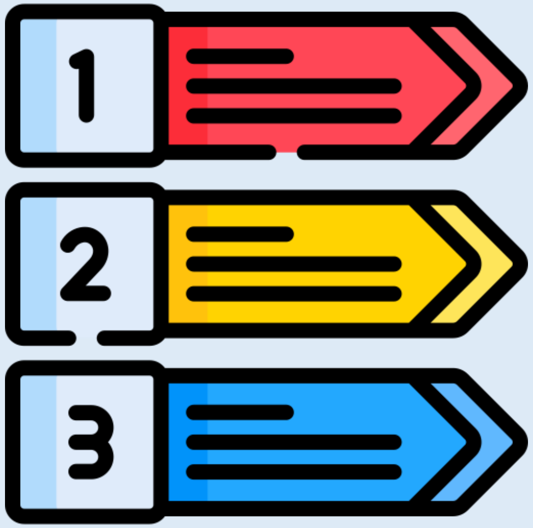
You can begin using S3 Object Lambda by following these simple steps:

- To transform data for your use case, create a Lambda Function.
- From the S3 Management Console, create an S3 Object Lambda Access Point.
- Choose the Lambda function you created earlier.
- Give S3 Object Lambda access to the original object by providing a supporting S3 Access Point.
- To retrieve data from S3, update your application configuration to use the new S3 Object Lambda Access Point.

.

Use Cases

Use Case of S3 object Lambda



- Personal identifiable information is redacted for analytics or non-production environments.
- Converting between data formats, such as XML to JSON.
- Adding information from other services or databases to data.
- As files are downloaded, they are compressed or decompressed.
- Using caller-specific details, such as the user who requested the object, resize and watermark images on the fly.
- Using custom authorization rules to gain access to data.

.

☎ India : +91-7847955955

US : 1-800-216-8930 (TOLL FREE)

✉ support@intellipaat.com

24/7 Chat with Our Course Advisor