AWS Foundation

Storage: Simple Storage Service (S3)

Agenda

1. Pre-S3: Online Cloud Storage
2. Pre-S3: API
3. S3 Introduction
4. S3 Consistency Models
5. Examples
6. Demo
7. Storage Hierarchy
8. Buckets
9. Demo
10. Objects
11. Metadata and Storage Class
12. Versioning
13. Demo
14. Lifecycle Management
15. Storage Class Analysis
16. Demo
17. Cross-region Replication
18. Data Encryption
19. Server Access Logging
20. Demo
21. Connecting Using VPC Endpoint
22. S3 Pricing

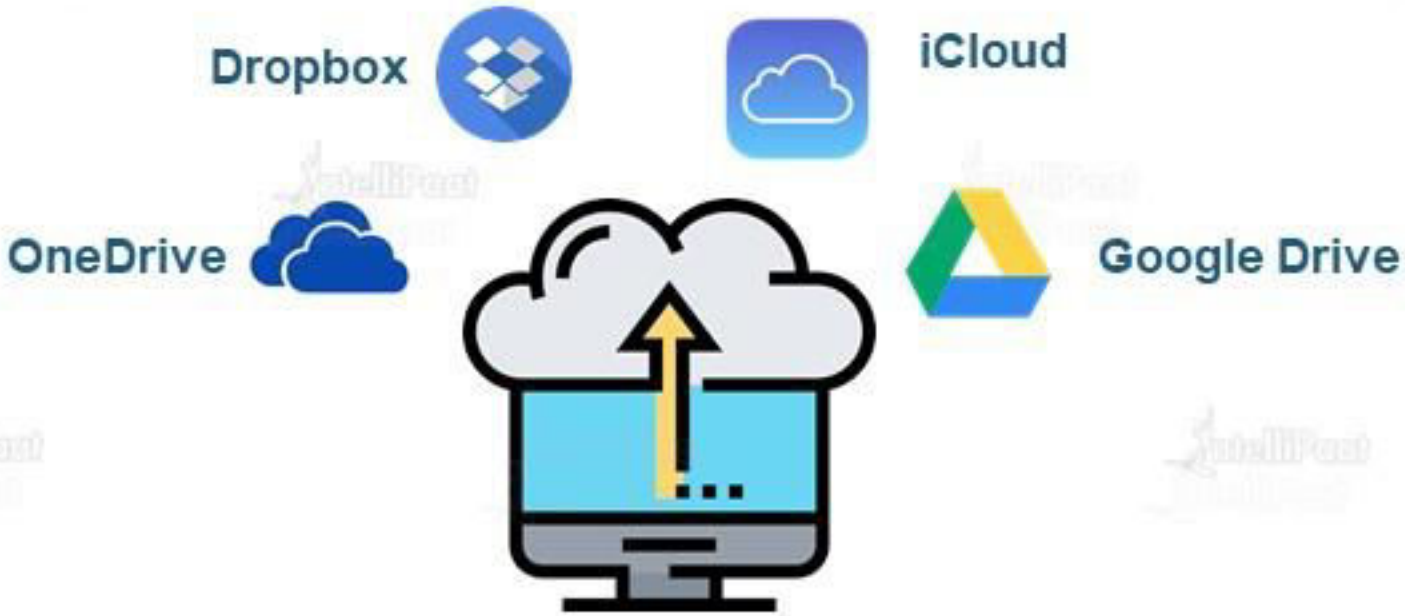
Copyright Intellipaat. All rights reserved.

Pre-S3: Online Cloud Storage

Copyright Intellipaat. All rights reserved.

Online Cloud Storage

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



Copyright Intellipaat. All rights reserved.

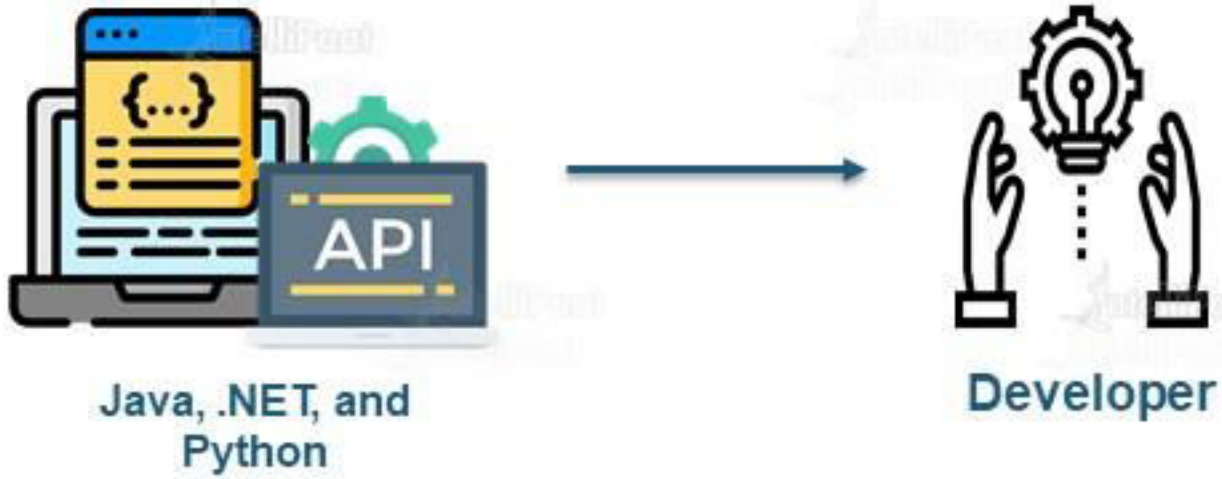
Pre-S3: API

Copyright Intellipaat. All rights reserved.

Application Programming Interface (API)

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



Copyright Intellipaat. All rights reserved.

S3 Introduction

Copyright Intellipaat. All rights reserved.

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

Copyright Intellipaat. All rights reserved.

S3 Consistency Models

Copyright Intellipaat. All rights reserved.

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

Copyright Intellipaat. All rights reserved.

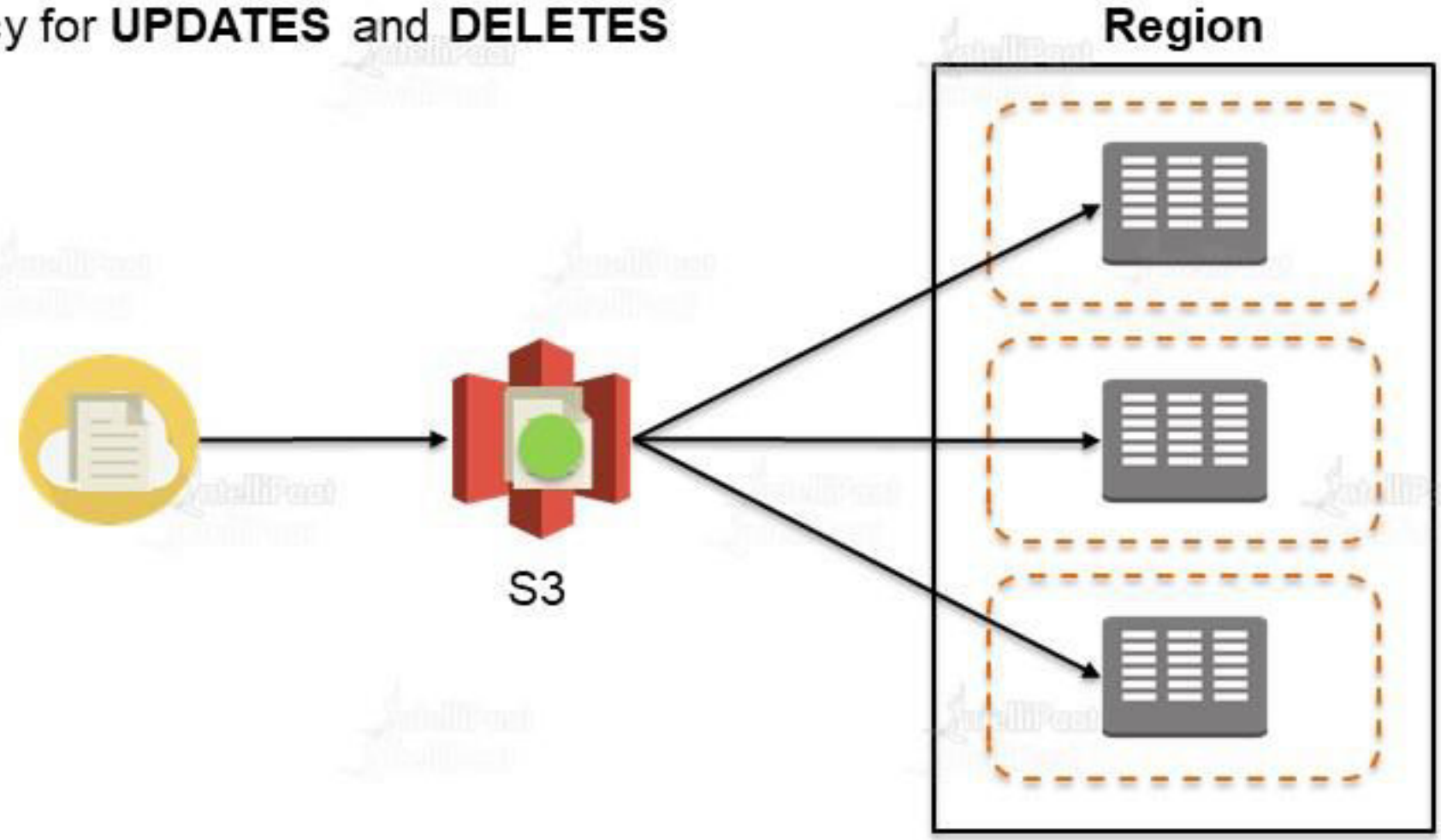
S3 Data Consistency Model

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

Copyright Intellipaat. All rights reserved.

Consistency Models

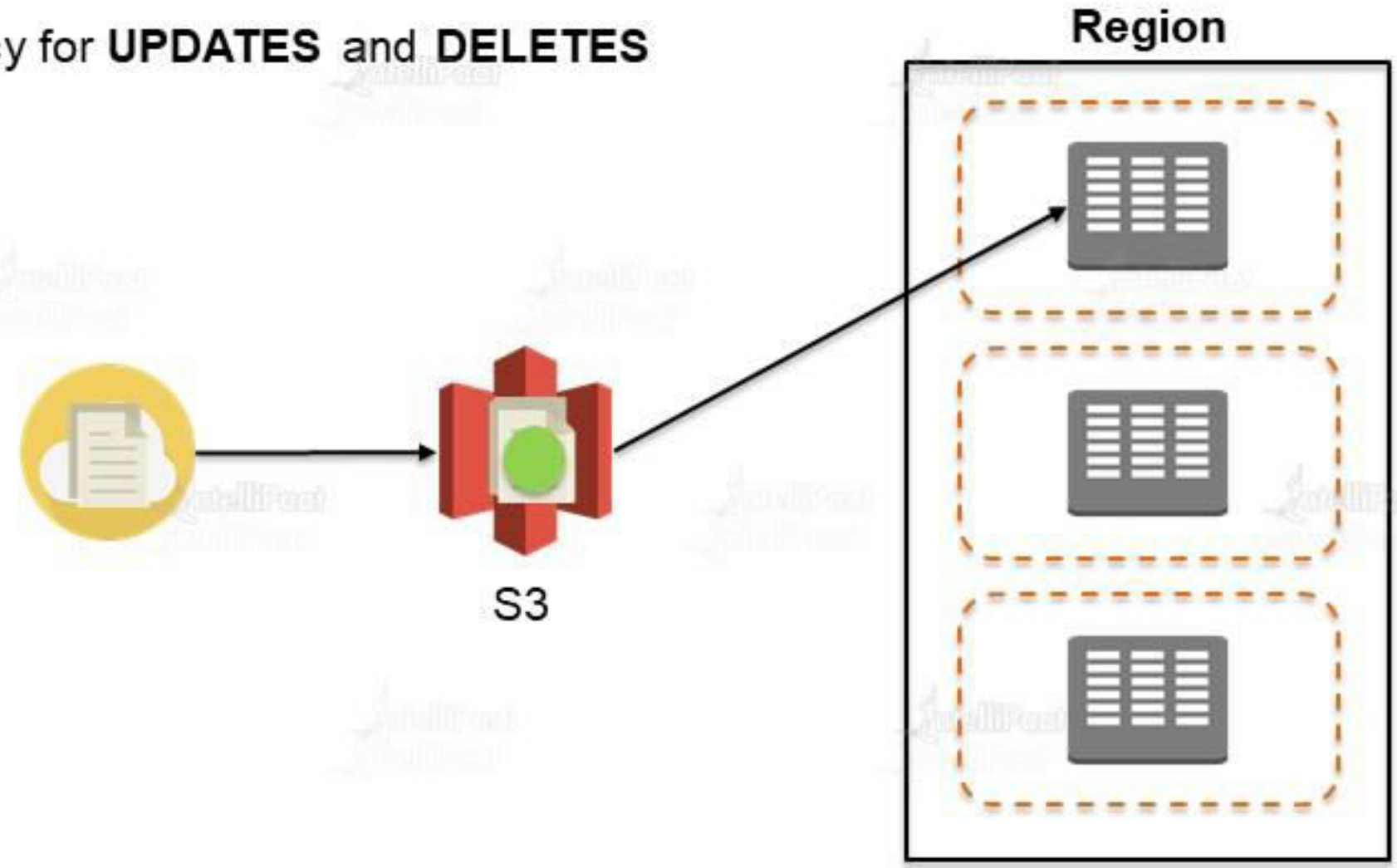
Eventual consistency for **UPDATES** and **DELETES**



Copyright Intellipaat. All rights reserved.

Consistency Models

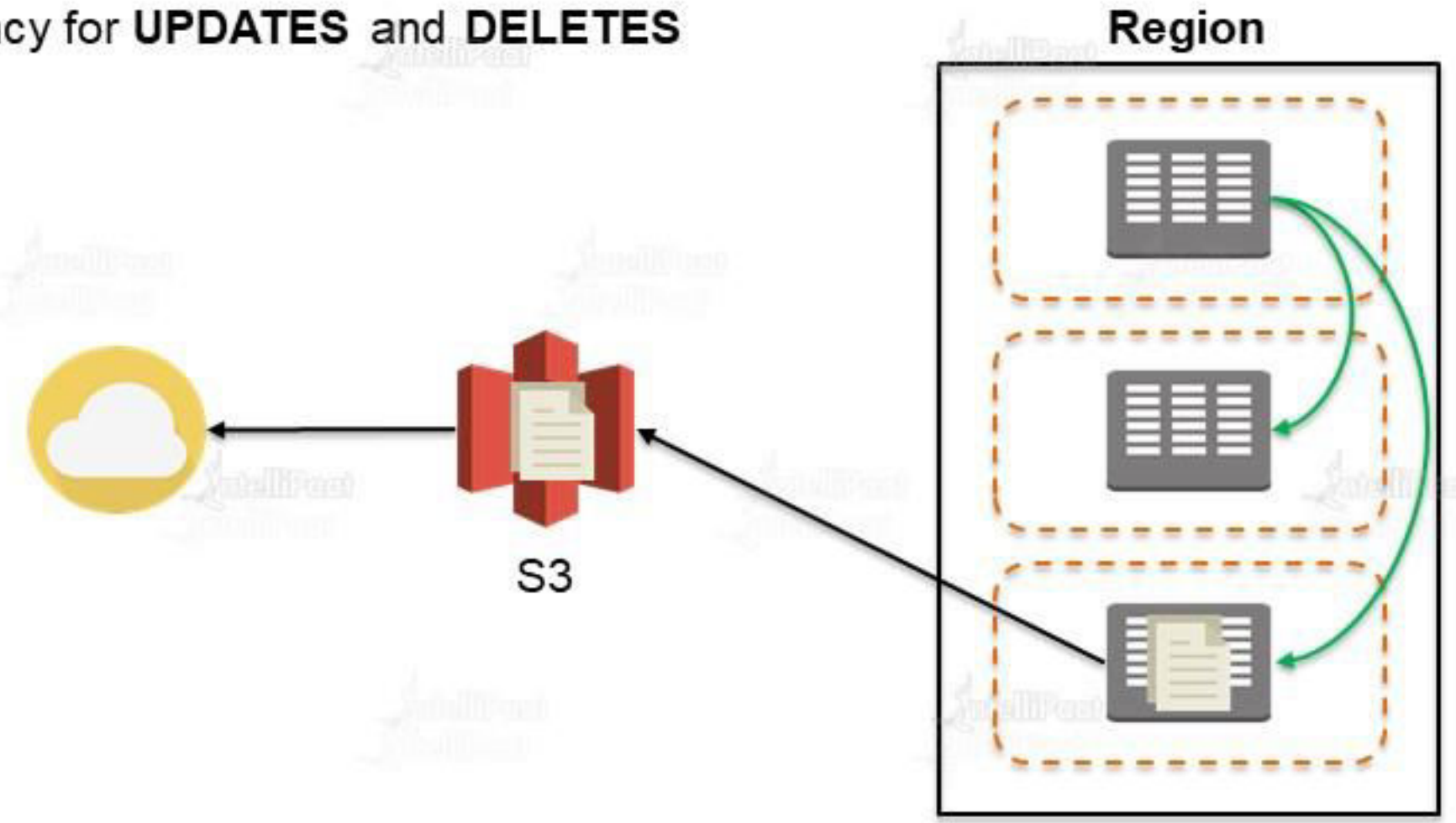
Eventual consistency for **UPDATES** and **DELETES**



Copyright Intellipaat. All rights reserved.

Consistency Models

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



Copyright Intellipaat. All rights reserved.

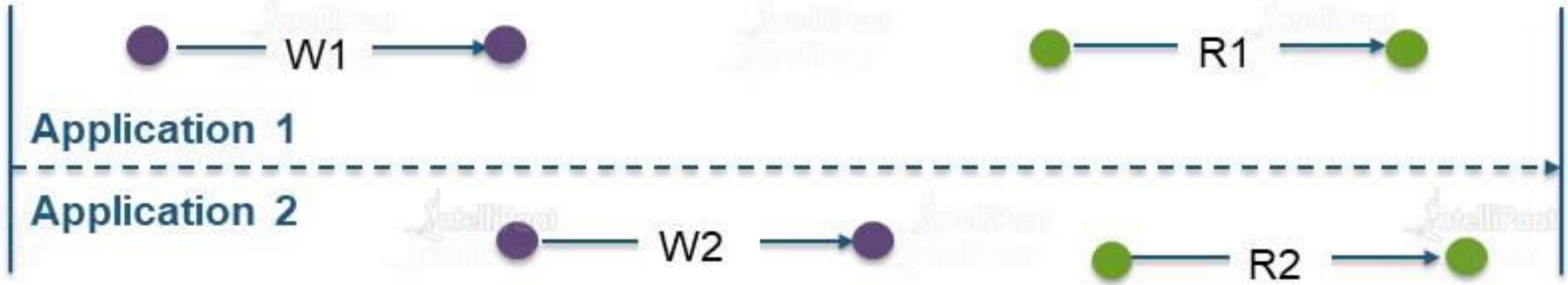
Examples

Copyright Intellipaat. All rights reserved.

Consistency Examples

Example 1

- Concurrent applications



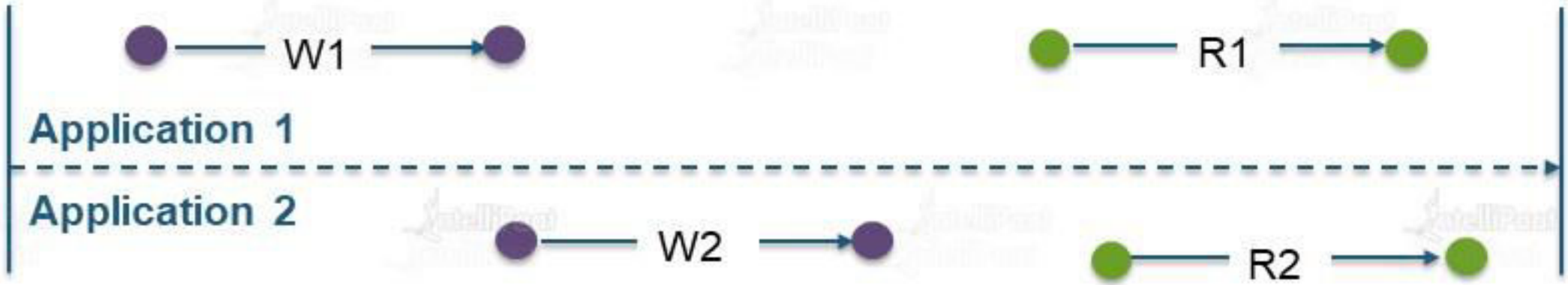
- W1 → Name: "EC2", W2 → Name: "EBS"
- R1 → consistent – Name: "EBS", eventual – Name: "EBS" or "EC2" or Nothing
- R2 → consistent – Name: "EBS", eventual – Name: "EBS" or "EC2" or Nothing

Copyright Intellipaat. All rights reserved.

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

Copyright Intellipaat. All rights reserved.

Demo

Copyright Intellipaat. All rights reserved.

Demo 1: Uploading Files in S3

- Let's upload a few image and text files in S3
- How to access those files

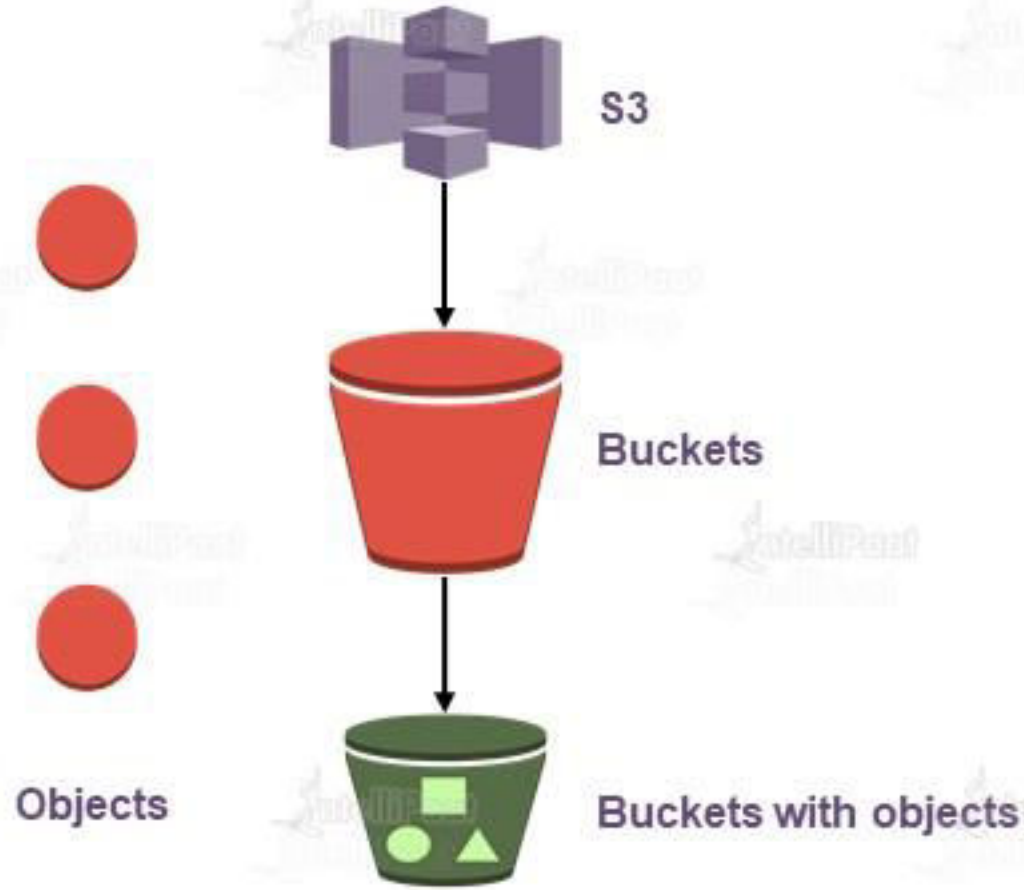
Copyright Intellipaat. All rights reserved.

Storage Hierarchy

Copyright Intellipaat. All rights reserved.

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



Copyright Intellipaat. All rights reserved.

Buckets

Copyright Intellipaat. All rights reserved.

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

Copyright Intellipaat. All rights reserved.

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
```

Copyright Intellipaat. All rights reserved.

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

Copyright Intellipaat. All rights reserved.

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

Copyright Intellipaat. All rights reserved.

Objects

Copyright Intellipaat. All rights reserved.

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)

Copyright Intellipaat. All rights reserved.

Metadata and Storage Class

Copyright Intellipaat. All rights reserved.

Metadata and Storage Class

- 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

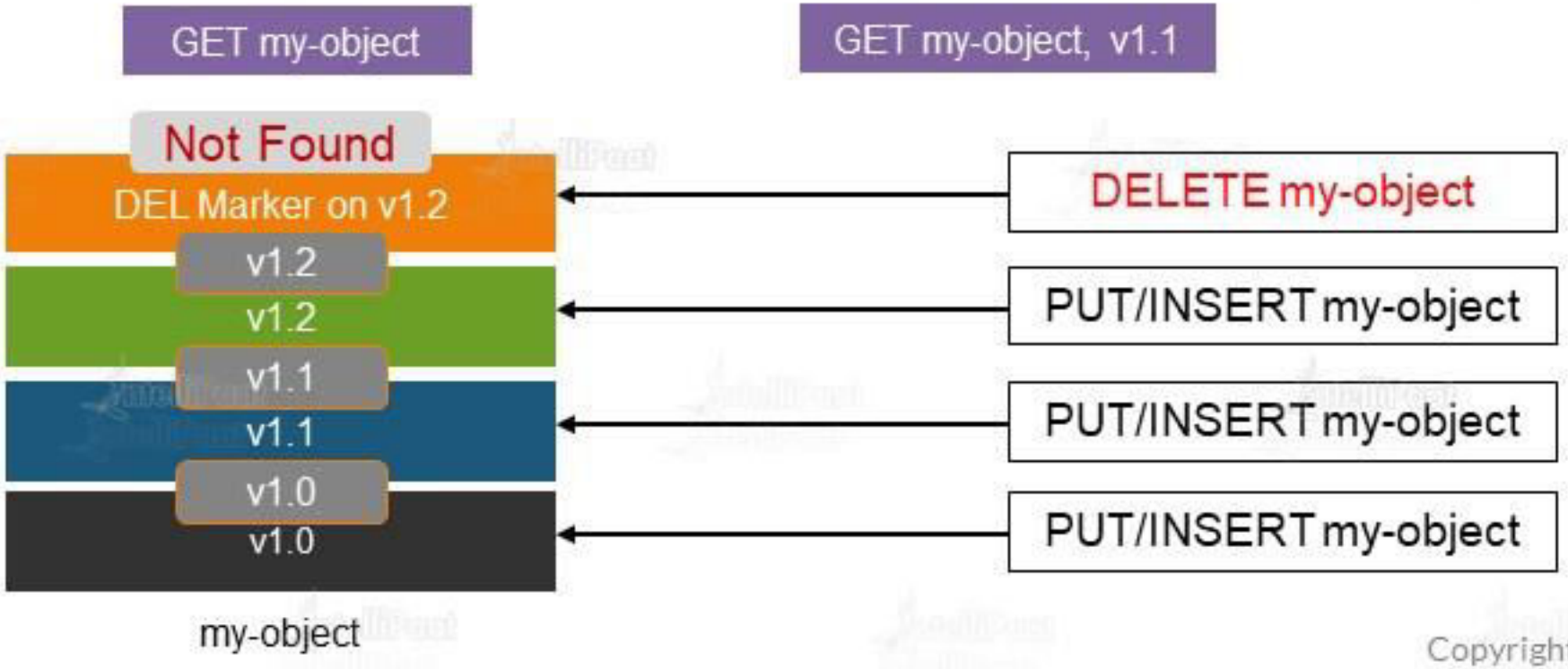
Copyright Intellipaat. All rights reserved.

Versioning

Copyright Intellipaat. All rights reserved.

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



Copyright Intellipaat. All rights reserved.

Demo

Copyright Intellipaat. All rights reserved.

Demo 2: Objects

- Upload a few objects in the bucket **aws-foundation-bucket** created earlier
- Set permission to everyone for both the bucket and the objects
- Check the storage class for the object created
- Enable versioning in the bucket
- Add **application:AWS-3** and **Content-type** metadata in one of the objects

Copyright Intellipaat. All rights reserved.

Lifecycle Management

Copyright Intellipaat. All rights reserved.

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

Copyright Intellipaat. All rights reserved.

Storage Class Analysis

Copyright Intellipaat. All rights reserved.

Storage Class Analysis

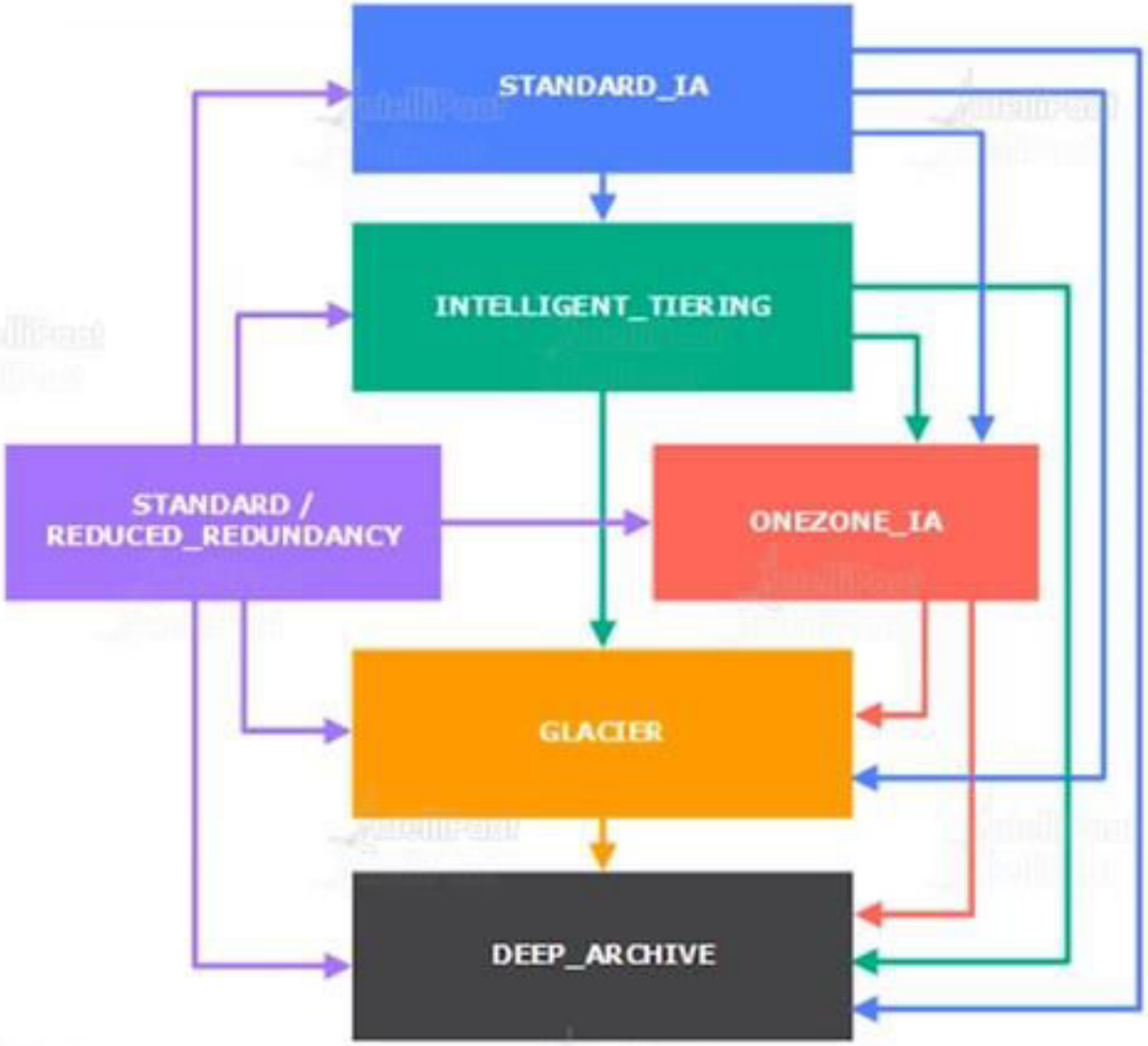
- Provides storage access patterns that can help us decide when the data/objects should be transitioned
- Maximum 1,000 storage class filtered analysis per bucket
- Analysis patterns:
  - Analyze the entire content of a bucket
  - Analyze objects grouped by tags or prefixes
- Storage class analysis observes 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

Copyright Intellipaat. All rights reserved.

Storage Class Analysis

Copyright Intellipaat. All rights reserved.

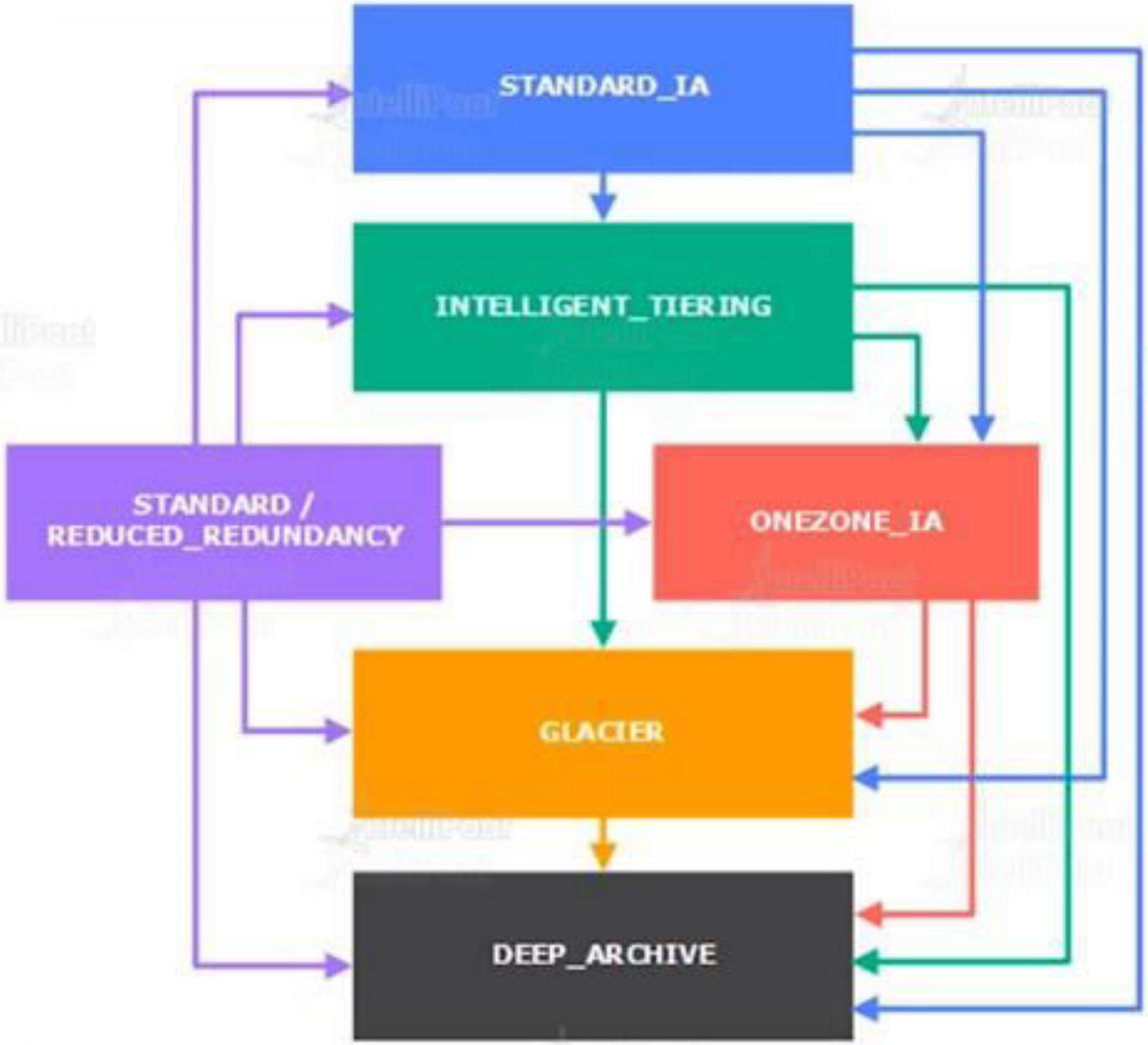
Storage Class Analysis



- 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

Copyright Intellipaat. All rights reserved.

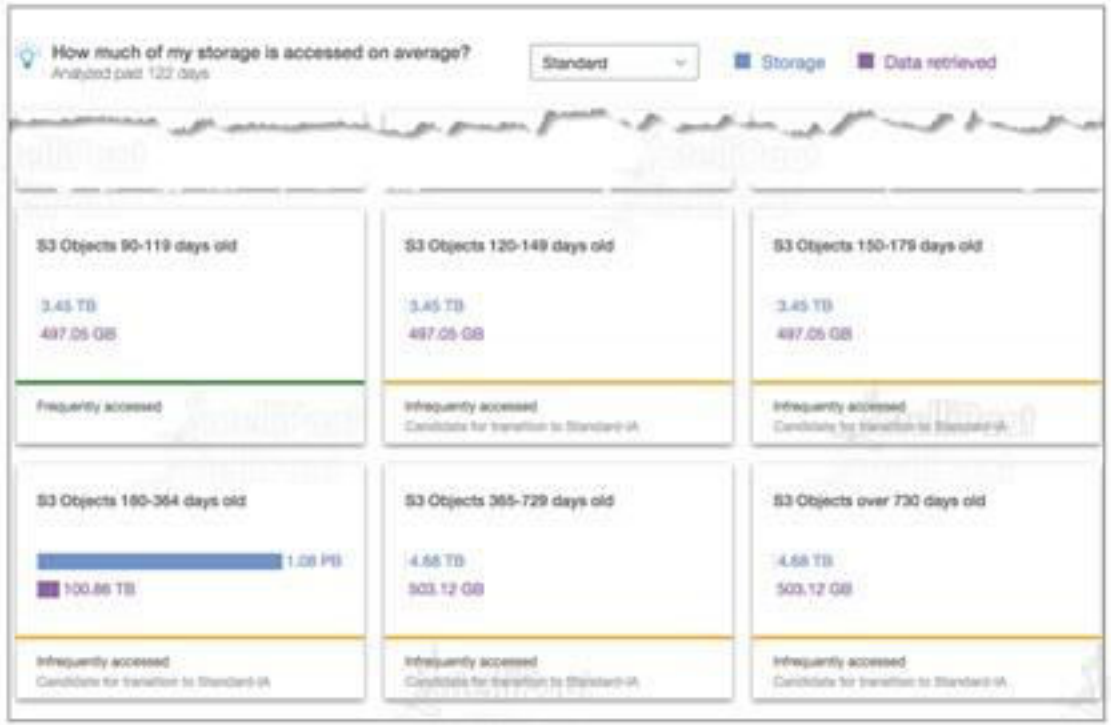
Storage Class Analysis



- 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

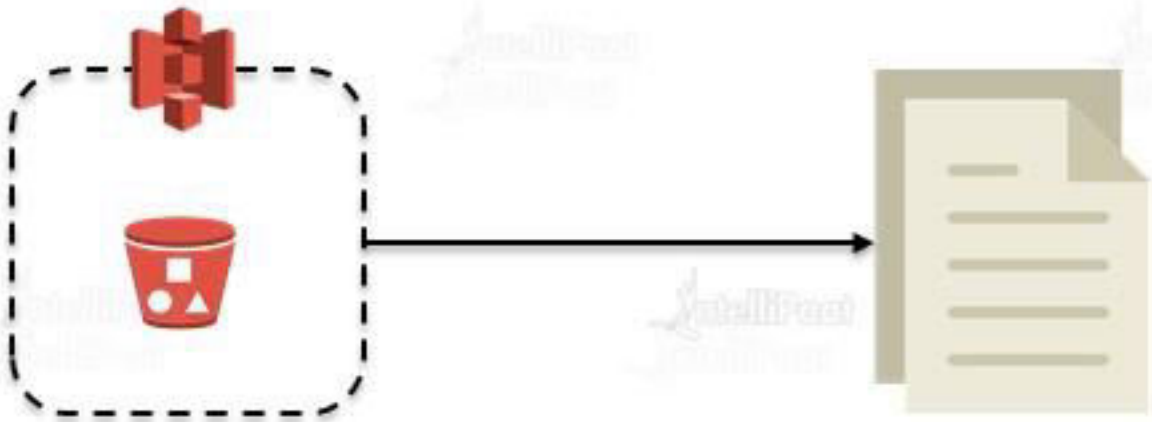
Copyright Intellipaat. All rights reserved.

Storage Class Analysis



Copyright Intellipaat. All rights reserved.

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

Copyright Intellipaat. All rights reserved.

Demo 3: Lifecycle Management

- Set up Lifecycle Management for only a few objects in a bucket
- Set up Lifecycle Management for objects with the tag ‘Training’

Copyright Intellipaat. All rights reserved.

Cross-region Replication

Copyright Intellipaat. All rights reserved.

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

Copyright Intellipaat. All rights reserved.

Cross-region Replication

- Compliance requirements
- Latency
- Operational
- Ownership

Copyright Intellipaat. All rights reserved.

Data Encryption

Copyright Intellipaat. All rights reserved.

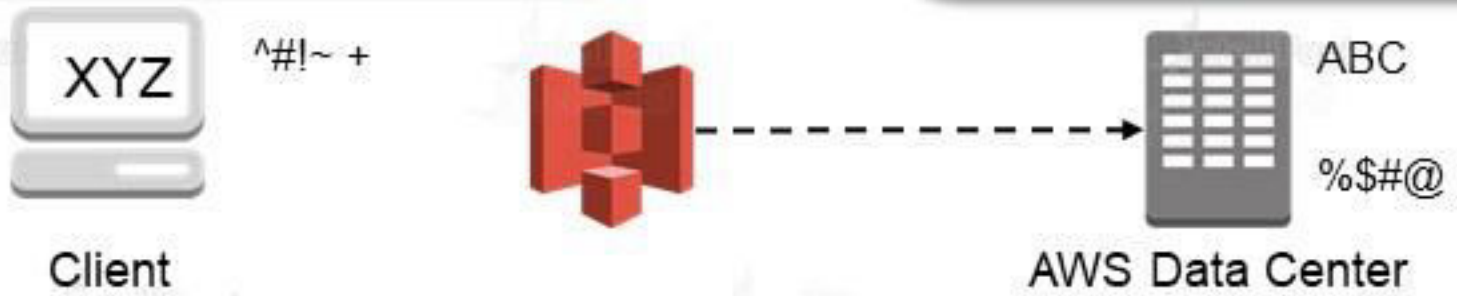
Data Encryption

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



Copyright Intellipaat. All rights reserved.

Server Access Logging

Copyright Intellipaat. All rights reserved.

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

Copyright Intellipaat. All rights reserved.

Demo 4: Website Hosting

- Create a bucket: <your-name>.s3.static.wesite.bucket (or whatever is suitable for your need)
- Upload your HTML page as an object
- Create sub-folder1 in the bucket
- Upload another HTML document in the sub-folder1. The name of this HTML document should be same as that of the one in the bucket
- Create sub-folder2 inside sub-folder1, and upload an HTML document with the same name as earlier
- Access the root page and all subpages using the S3 endpoint
- Create one more bucket, and make it to hold all logs for the bucket created in the first step
- Configure redirect so that requests to your main page are redirected to /sub-folder1/sub-folder2
- Integrate S3 website hosting with a custom domain name using Route 53

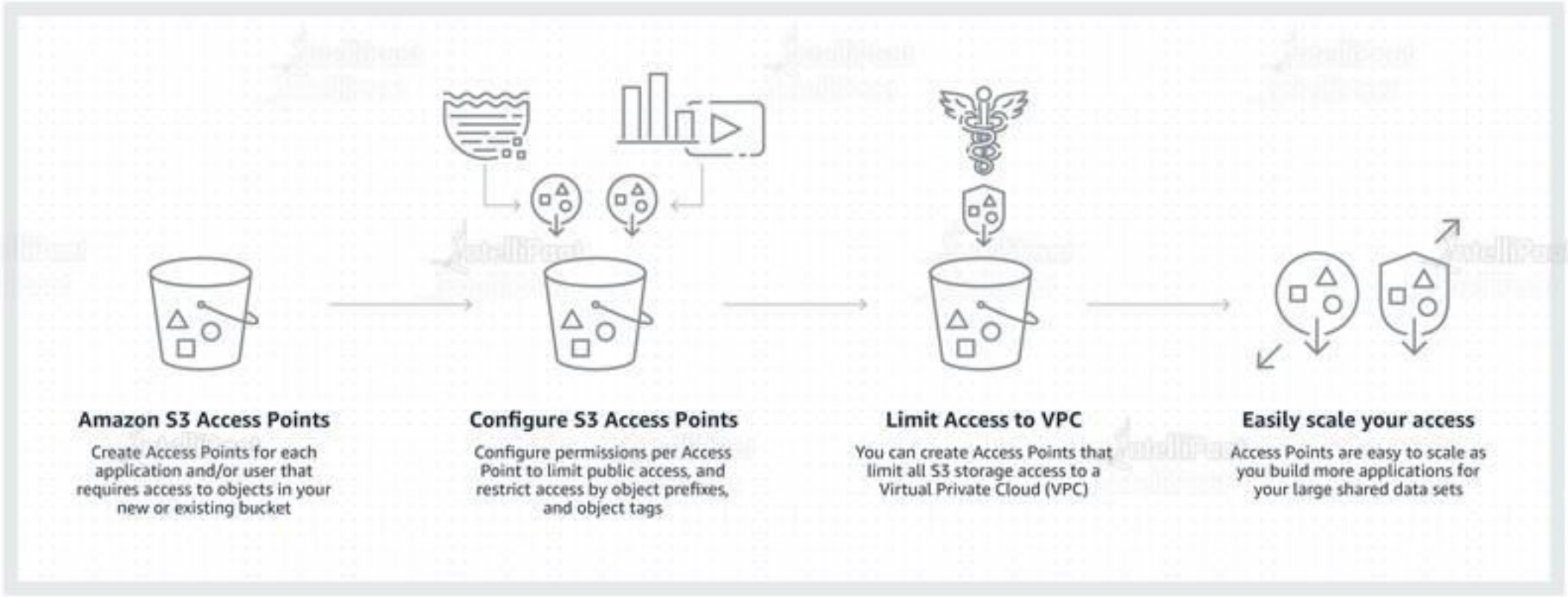
Copyright Intellipaat. All rights reserved.

S3 Access Points

Copyright Intellipaat. All rights reserved.

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



Copyright Intellipaat. All rights reserved.

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

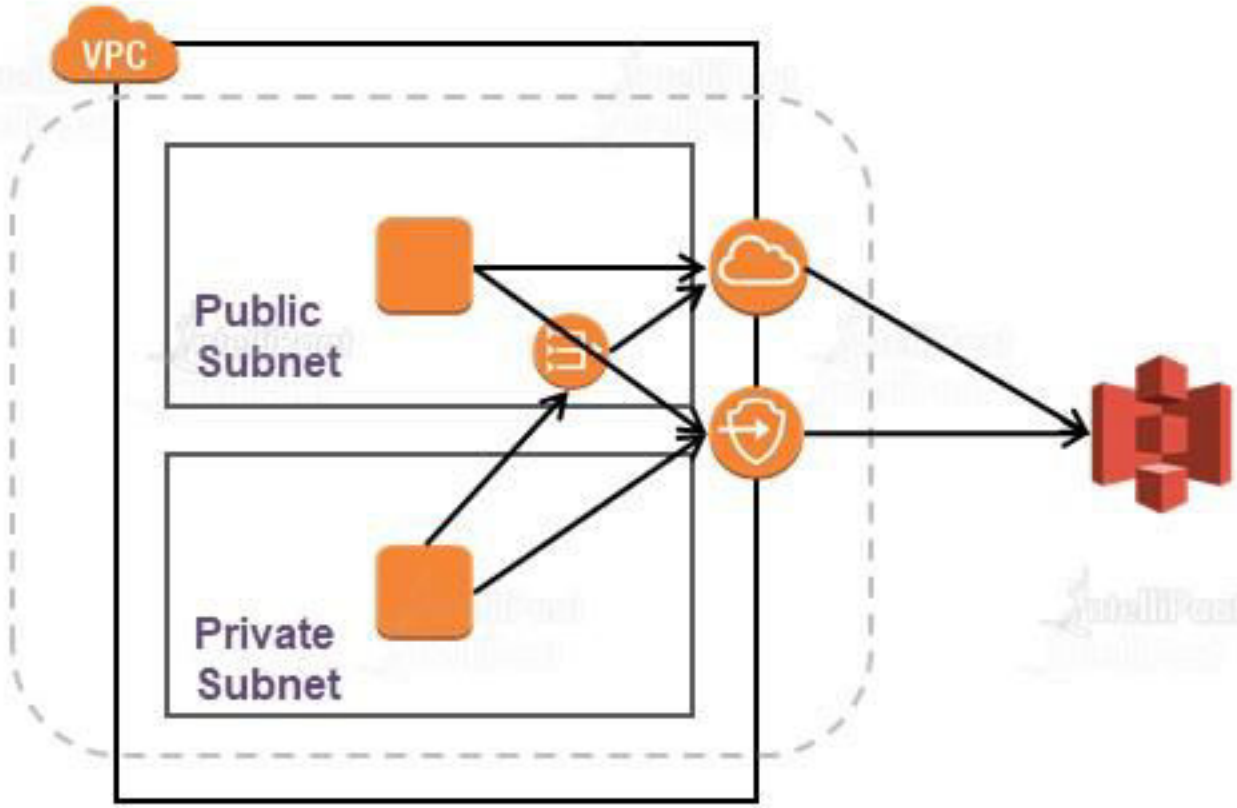
Copyright Intellipaat. All rights reserved.

Connecting Using VPC Endpoint

Copyright Intellipaat. All rights reserved.

Connecting Using VPC Endpoint

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



Copyright Intellipaat. All rights reserved.

Connecting Using VPC Endpoint

- 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

Copyright Intellipaat. All rights reserved.

S3 Pricing (us-east-1)

Copyright Intellipaat. All rights reserved.

S3 Pricing (us-east-1)

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

Copyright Intellipaat. All rights reserved.

IntelliPaat

India : +91-7847955955

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

sales@intellipaat.com

24/7 Chat with Our Course Advisor