

S3 Introduction:

## Section introduction

- Amazon S3 is one of the main building blocks of AWS
- It's advertised as "infinitely scaling" storage
- Many websites use Amazon S3 as a backbone
- Many AWS services use Amazon S3 as an integration as well
- We'll have a step-by-step approach to S3

1.

Sure. Amazon Simple Storage Service (S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance. Customers of all sizes and industries can store and protect any amount of data for virtually any use case, such as data lakes, cloud-native applications, and mobile apps.

Here are some of the key features of Amazon S3:

2.

- **Scalability:** Amazon S3 is highly scalable and can store any amount of data. You can start with a small amount of data and scale up as your needs grow.
- **Data availability:** Amazon S3 is designed to be highly available and durable. Your data is stored in multiple Availability Zones, so you can be confident that it will be available even if one Availability Zone fails.
- **Security:** Amazon S3 is designed to be secure. Your data is encrypted at rest and in transit. You can also use access control lists (ACLs) to control who has access to your data.
- **Performance:** Amazon S3 is designed to be performant. You can access your data from anywhere in the world with low latency.

3. Use cases for amazon s3..it used for backup and storage could be files or anything
4. Also used for disaster recovery if one region goes down ...we can move our data to other region
5. We can also archive files in S3 and later retrieve it for much cheaper usage
6. For hybrid cloud storage
7. Used for application hosting,media hosting like videos etc

## Amazon S3 Use cases

- Backup and storage
- Disaster Recovery
- Archive
- Hybrid Cloud storage
- Application hosting
- Media hosting
- Data lakes & big data analytics
- Software delivery
- Static website



Nasdaq stores 7 years of data into S3 Glacier



Sysco runs analytics on its data and gain business insights

- 8.
9. Nasdaq stores seven years of data into the S3 Glacier share service, which is like the archival service of Amazon S3.
10. And Sysco runs analytics on its data and gains business insights from Amazon S3.
11. Amazon S3-buckets
12. So Amazon S3 stores files into buckets. And buckets can be seen as top level directories.
13. The only thing that must be globally unique in AWS is amazon s3

## Amazon S3 - Buckets

- Amazon S3 allows people to store objects (files) in “buckets” (directories)
- Buckets must have a **globally unique name** (across all regions all accounts)
- Buckets are defined at the region level
- S3 looks like a global service but buckets are created in a region
- Naming convention
  - No uppercase, No underscore
  - 3-63 characters long
  - Not an IP
  - Must start with lowercase letter or number
  - Must NOT start with the prefix xn--
  - Must NOT end with the suffix -s3alias



S3 Bucket

14. \_\_\_\_\_

15. Amazon S3 Objects

16. The objects(files) have a key

## Amazon S3 - Objects

- Objects (files) have a Key
- The **key** is the FULL path:
  - s3://my-bucket/**my\_file.txt**
  - s3://my-bucket/**my\_folder1/another\_folder/my\_file.txt**
- The key is composed of **prefix** + **object name**
  - s3://my-bucket/**my\_folder1/another\_folder**/**my\_file.txt**
- There's no concept of “directories” within buckets (although the UI will trick you to think otherwise)
- Just keys with very long names that contain slashes (“/”)



Object



S3 Bucket  
with Object

17.

18. The key is composed of Path and object name

19. We can also nest our object in different folders ...

20. The values of the key are content of the object..

21. We can define our own metadata(list of key value pairs) or the system defines them

22. We have tags useful for security and lifecycle

# Amazon S3 – Objects (cont.)



- Object values are the content of the body:
    - Max. Object Size is 5TB (5000GB)
    - If uploading more than 5GB, must use “multi-part upload”
  - Metadata (list of text key / value pairs – system or user metadata)
  - Tags (Unicode key / value pair – up to 10) – useful for security / lifecycle
  - Version ID (if versioning is enabled)
- 23.

S3 Hands On:

1. Here first we will create a bucket..because this is where our data gets stored
2. And bucket names must be unique in the region..so we have to create a unique bucket name...
3. Our buckets will be created in only one region..which we choose

The screenshot shows the 'Create bucket' wizard. The 'General configuration' step is selected. It includes fields for 'Bucket name' (set to 'stephane-demo-s3'), 'AWS Region' (set to 'EU (Ireland) eu-west-1'), and a 'Copy settings from existing bucket - optional' section with a 'Choose bucket' button.

4. Then we create a bucket with all the default settings
5. Next we will go into our bucket and add objects..here we add a coffee image ..then click on upload

6. After uploading ..if we go into our object..we can see its overview
7. If click on open inside object overview ..we can see our uploaded objects
8. Inside the overview if we try to open our objects by its object url..we get access denied
9. But if open it thru open button...it add a presigned signature of ours inside the url and ot says that I'm the one who is opening it
10. We can also create folders and add our objects..and the destination of our object would be inside the created folder

## S3 Security Bucket Policy

1. Policies

# Amazon S3 – Security

- User-Based
    - IAM Policies – which API calls should be allowed for a specific user from IAM
  - Resource-Based
    - Bucket Policies – bucket wide rules from the S3 console - allows cross account
    - Object Access Control List (ACL) – finer grain (can be disabled)
- 2.
  3. Here in resource based the bucket policies will allow a specific user to come in or allow or allow a user from another account,
  4. this is called cross-account to access your S3 Buckets. This is also how we'll make our S3 Buckets public
  5. The most common way to use security in S3 bucket is to use bucket policies

# Amazon S3 – Security

- User-Based
  - IAM Policies – which API calls should be allowed for a specific user from IAM
- Resource-Based
  - Bucket Policies – bucket wide rules from the S3 console - allows cross account
  - Object Access Control List (ACL) – finer grain (can be disabled)
  - Bucket Access Control List (ACL) – less common (can be disabled)
- Note: an IAM principal can access an S3 object if
  - The user IAM permissions ALLOW it OR the resource policy ALLOWS it
  - AND there's no explicit DENY
- Encryption: encrypt objects in Amazon S3 using encryption keys

- 6.
7. S3 Bucket policies
8. Json based policies ...

## S3 Bucket Policies

- JSON based policies
  - Resources: buckets and objects
  - Effect: Allow / Deny

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "PublicRead",  
      "Effect": "Allow",  
      "Principal": "*",  
      "Action": [  
        "s3:GetObject"  
      ],  
      "Resource": [  
        "arn:aws:s3:::examplebucket/*"  
      ]  
    }  
  ]  
}
```

- 9.

```
,  
  "Resource": [  
    "arn:aws:s3:::examplebucket/*"
```

10. Here in this json document we have resource [
11. We can see that it applies to every objects in the bucket(\*)
12. Next we have effect it allows or deny the actions..here in this example ..the action we allowed is "s3:GetObject"...

13. Here in our example principle represents the account or the user...so we allowing every user with a star to access out bucket and retrieve object

## S3 Bucket Policies

- JSON based policies
  - Resources: buckets and objects
  - Effect: Allow / Deny
  - Actions: Set of API to Allow or Deny
  - Principal: The account or user to apply the policy to
- Use S3 bucket for policy to:
  - Grant public access to the bucket
  - Force objects to be encrypted at upload

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "PublicRead",  
      "Effect": "Allow",  
      "Principal": "*",  
      "Action": [  
        "s3:GetObject"  
      ],  
      "Resource": [  
        "arn:aws:s3:::examplebucket/*"  
      ]  
    }  
  ]  
}
```

14.

15. Example : public access

## Example: Public Access - Use Bucket Policy



S3 Bucket Policy  
Allows Public Access



Anonymous www website visitor

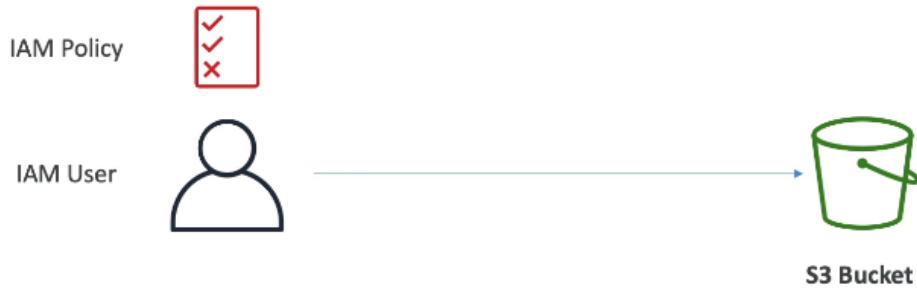


S3 Bucket

16.

17. Here in this example our s3 bucket has a policy attached to it and this allows public access...so anyone from internet can access our s3 bucket

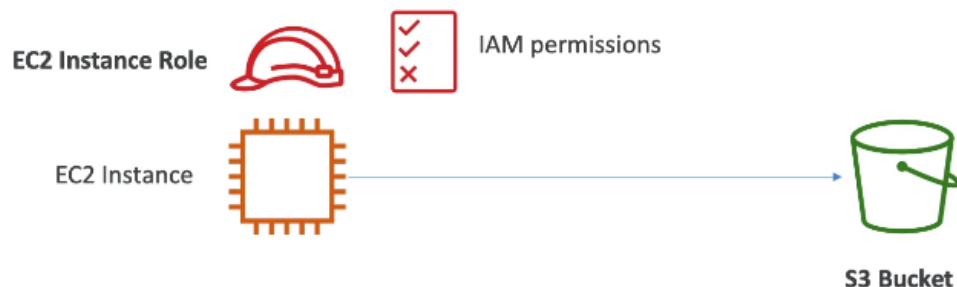
## Example: User Access to S3 – IAM permissions



18.

19. And if we have a user within our account which is IAM user ..we need to set IAM policy to the user to access S3 bucket

## Example: EC2 instance access - Use IAM Roles



20.

21. If we have ec2 instance and it wants to connect to s3 bucket...then we have create an ec2 instance role and assign correct IAM permissions to it to access s3 buckets

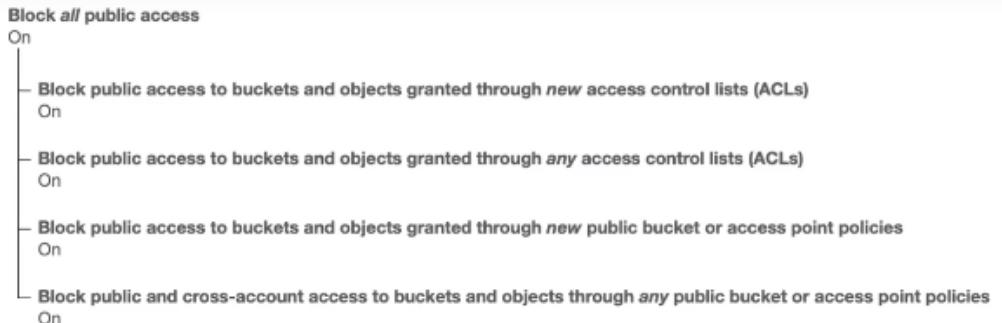
22.

## Advanced: Cross-Account Access – Use Bucket Policy



23. If we have IAM user in other AWS account ..then we create s3 bucket policy which allows cross account access for that specific IAM user..and therefore the IAM user will be able to make API calls into our S3 Buckets.

## Bucket settings for Block Public Access



- These settings were created to prevent company data leaks
- If you know your bucket should never be public, leave these on
- Can be set at the account level

24.

25. And if you know that none of your S3 Buckets ever should be public, you can set this at account level itself

## S3 Security policy Hands on

1. Here in our current bucket we have an object called coffee image...if we open it through its object url we get

```
▼<Error>
  <Code>AccessDenied</Code>
  <Message>Access Denied</Message>
  <RequestId>05G573T9BC17T411</RequestId>
  <HostId>5P1ZGfcM/pX7acUjnQDnG1TcFj7AvmZVuJH93BFUV4wy7EgaCeDkGNrcrVM3ZInhVaQFd0t9023DZRhn3gba9yjZudGBfGGSMEMHo8Gjws=</HostId>
</Error>
```

2. So to allow it for public ..we have to go to permissions tab..and we allow block public access to allow it for public
3. We allowed public access

### Block all public access

 Off

► Individual Block Public Access settings for this bucket

- 
4. Next we go to the bucket policy..currently we don't have any bucket policy
  5. Now we create one for it ..for that we use policy generator..in the policy generator..
  6. We select the s3 bucket policy and in principle we keep "\*" to allow all..in actions we select "getObjects" to read the bucket objects
  7. In ARN we give our bucket arn..here we add "/" And the reason why we do this is
  8. that this action the GetObject action right here applies to objects within your buckets and therefore objects within your buckets are after "/" and their "\*" to represent these objects.
  9. Now we have to paste this generated policy into our bucket policy

### Permissions overview

Access

 Public

10. Now after saving we can see our access due to our generated policy

**Permissions overview**

Access

**⚠️ Public**

**Publicly accessible** X

This bucket can be accessed by anyone on the internet. Unless you require a public configuration for a specific verified use case, AWS recommends that you block all public access to your buckets. [Learn more](#)

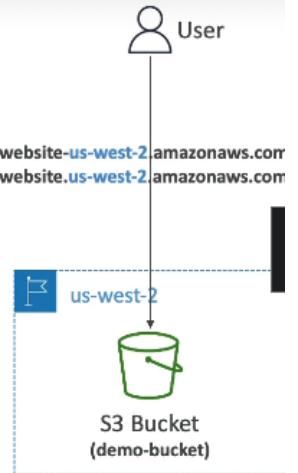
**Block public access**

- 11.
12. Now if refresh the our object url...we can see the coffee image..it means our bucket object has been made public
13. And any object which are in our bucket..can be accessed by public by its object url

### Amazon S3 - Static Website Hosting

## Amazon S3 – Static Website Hosting

- S3 can host static websites and have them accessible on the Internet
- The website URL will be (depending on the region)
  - [http://bucket-name.s3-website-aws-region.amazonaws.com](http://bucket-name.s3-website-us-west-2.amazonaws.com)
  - OR
  - [http://bucket-name.s3-website.aws-region.amazonaws.com](http://bucket-name.s3-website.us-west-2.amazonaws.com)
- If you get a 403 Forbidden error, make sure the bucket policy allows public reads!



- 1.
2. Lets consider our bucket has some html files,images etc and a user can access the url ..and if he gets an 403 forbidden error...then see pic

### S3 Website Hands on

1. Now lets enable our bucket to be a website...
2. Here we added a beach file to our bucket.
3. If we go to properties tab of our bucket...at bottom we have static website hosting..we have to enable it

Index document  
Specify the home or default page of the website.

[index.html](#)

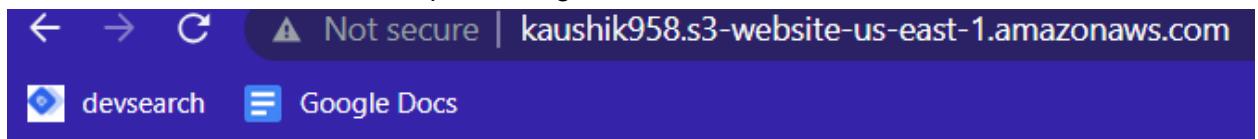
4. Here we have to add index.html file
5. Next we will upload index.html file to bucket
6. Now again go to static website hosting..we have bucket website endpoint

Bucket website endpoint

When you configure your bucket as a static website, the website is available at the AWS Region-specific website endpoint of the bucket. [Learn more](#)

<http://stephane-demo-s3.s3-website-eu-west-1.amazonaws.com>

7. If we click on bucket website endpoint..we get



## I love coffee

Hello world!

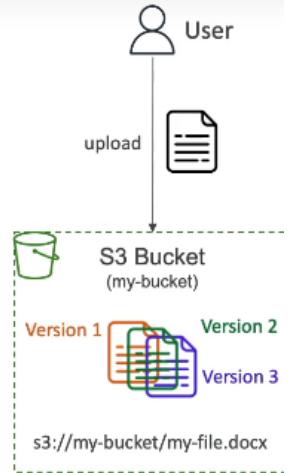


8. And if we open our image on new tab...we get our image's object url

## S3 Versioning overview

# Amazon S3 - Versioning

- You can version your files in Amazon S3
- It is enabled at the bucket level
- Same key overwrite will change the “version”: 1, 2, 3....
- It is best practice to version your buckets
  - Protect against unintended deletes (ability to restore a version)
  - Easy roll back to previous version
- Notes:
  - Any file that is not versioned prior to enabling versioning will have version “null”
  - Suspending versioning does not delete the previous versions



- 1.
2. Here lets suppose a user uploads a file to our bucket...if versioning is enabled..
3. S3 bucket will create a version to the uploaded file at the selected key..so when a user re-uploads the same key..to overwrite the current file...it creates a version two..then version 3..so on ..just like github

## S3 Versioning Hands On:

1. First go into properties..there we have bucket versioning..we have to click edit and enable it

- Now in the bucket we have index.html file...now lets update our file in our local device and reupload it again..now we get I love tea

## I love Tea

Hello world!



- In last version it was "I love Coffee"
- If we click on show versions...we get

<input type="checkbox"/>	Name	Type	Version ID	Last modified	Size	Storage class
<input type="checkbox"/>	beach.jpg	jpg	null	October 11, 2022, 11:08:32 (UTC+01:00)	85.8 KB	Standard
<input type="checkbox"/>	coffee.jpg	jpg	null	October 11, 2022, 10:32:56 (UTC+01:00)	108.4 KB	Standard
<input type="checkbox"/>	index.html	html	xqkPZwl095FfCogWMXVWwH1oWv0Fb_4P	October 11, 2022, 11:27:52 (UTC+01:00)	207.0 B	Standard
<input type="checkbox"/>	index.html	html	<del>null</del>	October 11, 2022, 11:11:46 (UTC+01:00)	200.0 B	Standard

- Here we have uploaded 3 files before enabling versioning so their id's set to null
- Here we can see we have version id to the file index.html that we uploaded now
- And after if want to see our old index.html..file..we have to delete the new index.html
- If we delete a specific version ID of our objects ...it permanently deletes them
- And if we disable our show versions ..it adds a delete marker to them
- Basically here..if we enable show version and delete a file..it permanently deletes the file
- If we disable the show version and deletes and file..it adds a delete marker ..but in reality it is still there in our bucket..we can see that by enabling versions
- Suppose if we delete an object ..then it will create delete marker...
- Now to restore that object..we have enable version and delete the delete marker...then it will restore our object again

## Amazon S3 - Replication(CRR & SRR)

- So CRR is for cross-region replication and SRR is for same-region replication.
- Here we have 2 buckets in two regions ..and we want to setup asynchronous replication

3. What is asynchronous replication?

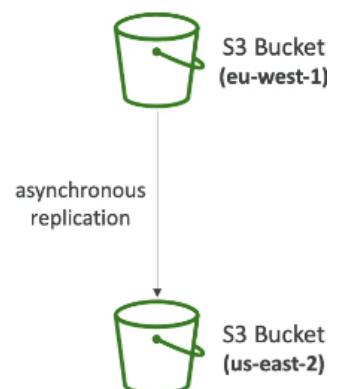
Asynchronous replication in AWS is a data storage backup technique where data is not immediately backed up during or immediately after the primary storage acknowledges write complete, but rather done over a period of time. This type of replication is often used to achieve high availability and disaster recovery (DR) in AWS.

4. To setup...we have to enable versioning

## Amazon S3 – Replication (CRR & SRR)



- Must enable Versioning in source and destination buckets
- Cross-Region Replication (CRR)
- Same-Region Replication (SRR)



5.

6. For CRR..two buckets must be in different region..for SRR must be in same region

- Buckets can be in different AWS accounts
  - Copying is asynchronous
  - Must give proper IAM permissions to S3
- 
- Use cases:
    - CRR – compliance, lower latency access, replication across accounts
    - SRR – log aggregation, live replication between production and test accounts

7. \_\_\_\_\_

8. Copying is asynchronous means it happens in the background

## S3 Replication Hands on

1. For this we need 2 buckets so..we create a new bucket with different region..then we also need to enable versioning ...because replication happens only if we enable versioning
2. Then we create a target bucket

<input type="radio"/>	kaushik958-origin	US East (Ohio) us-east-2	Bucket and objects not public	June 26, 2023, 21:05:10 (UTC-05:00)
3.	<input type="radio"/> kaushik958-target	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	June 26, 2023, 21:05:56 (UTC-05:00)

4. Now as you see ..we have created origin and target buckets
5. Then we upload files into origin bucket...here we have uploaded beach.jpg to our origin bucket
6. Now go to the management section of origin bucket..here if we scroll down we have there are replication rules
7. Now we will create demo replication rule...we have create demo rule and added destination bucket that is our target bucket which is in diff region..so that is CRR

**Destination**

**Destination**  
You can replicate objects across buckets in different AWS Regions (Cross-Region Replication) or you can replicate objects across buckets in the same AWS Region (Same-Region Replication). You can also specify a different bucket for each rule in the configuration. [Learn more](#)  or see [Amazon S3 pricing](#) 

Choose a bucket in this account  
 Specify a bucket in another account

**Bucket name**  
Choose the bucket that will receive replicated objects.  
 [Browse S3](#)

**Destination Region**  
8. US East (N. Virginia) us-east-1

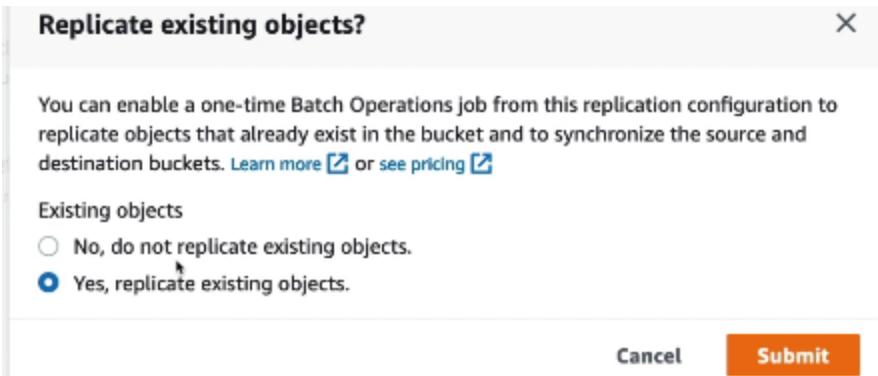
9. Now we choose IAM role as Create new role ..And then save

**IAM role**

Choose from existing IAM roles  
 Enter IAM role ARN

**IAM role**  
[Create new role](#)   

10. If we want to replicate existing objects then we have to enable one time batch see pic



- 11.
12. Here we didn't replicate existing objects
13. Now in the origin bucket ..if we upload a new file ..then the new file will be replicated in our target bucket
14. And the version id of this object will be same in both origin and target buckets

## S3 Storage Classes

1. Different types of storages in S3

## S3 Storage Classes

- Amazon S3 Standard - General Purpose
- Amazon S3 Standard-Infrequent Access (IA)
- Amazon S3 One Zone-Infrequent Access
- Amazon S3 Glacier Instant Retrieval
- Amazon S3 Glacier Flexible Retrieval
- Amazon S3 Glacier Deep Archive
- Amazon S3 Intelligent Tiering

- 2.
3. When we create an object in s3..we can choose its class and also we can modify class manually
4. S3 Durability and Availability
5. So durability represents how many times an object is going to be lost by Amazon S3...And so Amazon S3 has a very high durability.

# S3 Durability and Availability

- Durability:

- High durability (99.99999999%, 11 9's) of objects across multiple AZ
- If you store 10,000,000 objects with Amazon S3, you can on average expect to incur a loss of a single object once every 10,000 years
- Same for all storage classes

- Availability:

- Measures how readily available a service is
- Varies depending on storage class
- Example: S3 standard has 99.99% availability = not available 53 minutes a year

6.

7. Availability is measure of how readily a servie is...

8. Not available 53 minutes is may be due to some errors when we deal with the service

9. S3- standard-general purpose

## S3 Standard – General Purpose

10.

- 99.99% Availability
  - Used for frequently accessed data
  - Low latency and high throughput
  - Sustain 2 concurrent facility failures
- 
- Use Cases: Big Data analytics, mobile & gaming applications, content distribution...

Concurrent facility failure in Amazon S3 storage is a scenario in which two or more Availability Zones (AZs) in the same AWS Region fail at the same time. This could happen due to a natural disaster, a power outage, or a hardware failure.

Amazon S3 is designed to withstand the concurrent failure of two AZs in the same Region. This is achieved by replicating data across multiple AZs within a Region. In the event of a concurrent facility failure, Amazon S3 will be able to restore your data from the remaining AZs.

11.

12. S3 Infrequent Access

## S3 Storage Classes – Infrequent Access

- For data that is less frequently accessed, but requires rapid access when needed
- Lower cost than S3 Standard
- Amazon S3 Standard-Infrequent Access (S3 Standard-IA)
  - 99.9% Availability
  - Use cases: Disaster Recovery, backups



- Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA)
  - High durability (99.99999999%) in a single AZ; data lost when AZ is destroyed
  - 99.5% Availability



13.     • Use Cases: Storing secondary backup copies of on-premise data, or data you can recreate
14. We may have cost on retrieval for retrieving infrequent access data
15. S3 Glacier Storage Classess

# Amazon S3 Glacier Storage Classes

- Low-cost object storage meant for archiving / backup
- Pricing: price for storage + object retrieval cost
- **Amazon S3 Glacier Instant Retrieval**
  - Millisecond retrieval, great for data accessed once a quarter
  - Minimum storage duration of 90 days
- **Amazon S3 Glacier Flexible Retrieval (formerly Amazon S3 Glacier):**
  - Expedited (1 to 5 minutes), Standard (3 to 5 hours), Bulk (5 to 12 hours) – free
  - Minimum storage duration of 90 days
- **Amazon S3 Glacier Deep Archive – for long term storage:**
  - Standard (12 hours), Bulk (48 hours)
  - Minimum storage duration of 180 days

16.

17. Here in glacier Instant retrieval the data can be retrieved in milli seconds

18. IN flexible ..we have 3 options to retrieve and cost varies like expedited costs more than standard and standard costs more than bulk

19. In s3 deep archive data can be retrieved in 12hrs and 48hrs see pic

1. **Expedited Tier:** The Expedited Tier is the fastest retrieval option but also the most expensive. It allows you to retrieve data within 1-5 minutes. This tier is suitable for situations where you need immediate access to your archived data. However, keep in mind that you pay a higher price per gigabyte compared to other tiers.
  2. **Standard Tier:** The Standard Tier is the default retrieval option for Amazon S3 Glacier. It provides access to your data within 3-5 hours. This tier offers a balance between cost and retrieval speed. It is suitable for most use cases where you don't require immediate access to your data but still need it in a reasonably short time frame.
  3. **Bulk Tier:** The Bulk Tier is the most cost-effective retrieval option in Amazon S3 Glacier. It provides access to your data within 5-12 hours. This tier is suitable for scenarios where you have lower urgency for data retrieval and cost optimization is a priority. It offers the lowest price per gigabyte but has the longest retrieval time.
20. S3 Intelligent-tiering

# S3 Intelligent-Tiering

- Small monthly monitoring and auto-tiering fee
  - Moves objects automatically between Access Tiers based on usage
  - There are no retrieval charges in S3 Intelligent-Tiering
- 
- *Frequent Access tier (automatic)*: default tier
  - *Infrequent Access tier (automatic)*: objects not accessed for 30 days
  - *Archive Instant Access tier (automatic)*: objects not accessed for 90 days
  - *Archive Access tier (optional)*: configurable from 90 days to 700+ days
  - *Deep Archive Access tier (optional)*: config. from 180 days to 700+ days

22.

23. So S three Intelligent-Tiering is really to allow you to just sit back and relax while S three moves objects for you.

## S3 Storage Classes Comparison

	Standard	Intelligent-Tiering	Standard-IA	One Zone-IA	Glacier Instant Retrieval	Glacier Flexible Retrieval	Glacier Deep Archive
Durability	99.999999999% == (11 9's)						
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
Min. Storage Duration Charge	None	None	30 Days	30 Days	90 Days	90 Days	180 Days
Min. Billable Object Size	None	None	128 KB	128 KB	128 KB	40 KB	40 KB
Retrieval Fee	None	None	Per GB retrieved	Per GB retrieved	Per GB retrieved	Per GB retrieved	Per GB retrieved

24.

## S3 Storage Classes – Price Comparison

### Example: us-east-1

	Standard	Intelligent-Tiering	Standard-IA	One Zone-IA	Glacier Instant Retrieval	Glacier Flexible Retrieval	Glacier Deep Archive
Storage Cost (per GB per month)	\$0.023	\$0.0025 - \$0.023	%0.0125	\$0.01	\$0.004	\$0.0036	\$0.00099
Retrieval Cost (per 1000 request)	GET: \$0.0004 POST: \$0.005	GET: \$0.0004 POST: \$0.005	GET: \$0.001 POST: \$0.01	GET: \$0.001 POST: \$0.01	GET: \$0.01 POST: \$0.02	GET: \$0.0004 POST: \$0.03  Expedited: \$10 Standard: \$0.05 Bulk: free	GET: \$0.0004 POST: \$0.05  Standard: \$0.10 Bulk: \$0.025
Retrieval Time	Instantaneous					Expedited (1 – 5 mins) Standard (3 – 5 hours) Bulk (5 – 12 hours)	Standard (12 hours) Bulk (48 hours)
Monitoring Cost (per 1000 objects)		\$0.0025					

25.

### S3 Storage Hands on

1. First we will create a bucket with name s3-storageclassdemo bucket...next we will upload some files in the bucket
2. While uploading the file...we can look at the properties of the object..then we can see storage class..

Storage class	Designed for	Availability Zones	Min storage duration	Cost
<input checked="" type="radio"/> Standard	Frequently accessed data (more than once a month) with milliseconds access	≥ 3	-	-
<input type="radio"/> Intelligent-Tiering	Data with changing or unknown access patterns	≥ 3	-	-
<input type="radio"/> Standard-IA	Infrequently accessed data (once a month) with milliseconds access	≥ 3	30 days	1
<input type="radio"/> One Zone-IA	Recreatable, infrequently accessed data (once a month) stored in a single Availability Zone with milliseconds access	1	30 days	1
<input type="radio"/> Glacier Instant Retrieval	Long-lived archive data accessed once a quarter with instant retrieval in milliseconds	≥ 3	90 days	1
<input type="radio"/> Glacier Flexible Retrieval (formerly Glacier)	Long-lived archive data accessed once a year with retrieval of minutes to hours	≥ 3	90 days	-
<input type="radio"/> Glacier Deep	Long-lived archive data accessed less than once a year	≥ 3	180 days	-

- 3.
4. In one zone-IA ..we may be at risk of losing the data..like if any disaster occurs to our A-Z
5. Here in our example first we choose Standard-IA class to store our data ..
6. Later we can also change the storage class...of our object
7. We can also automate moving these objects between the different storage classes
8. For that we have to go to the buckets management section and create life cycles
9. In life cycle..we will create a rule and we will select action as moving current version of objects bw storage classes
10. Later we can add transitions ..like in how many days after it has to move to a diff storage class

## Create lifecycle rule Info

### Lifecycle rule configuration

Lifecycle rule name

DemoRule

Up to 255 characters

Choose a rule scope

- Limit the scope of this rule using one or more filters  
 Apply to all objects in the bucket

11.

### Lifecycle rule actions

Choose the actions you want this rule to perform. Per-request fees apply. [Learn more](#)  or see [Amazon S3 pricing](#) 

12.

- Move current versions of objects between storage classes

#### Transition current versions of objects between storage classes

Choose transitions to move current versions of objects between storage classes based on your use case scenario and performance access requirements. These transitions start from when the objects are created and are consecutively applied. [Learn more](#) 

Choose storage class transitions

Standard-IA

Days after object creation

30

Remove

Intelligent-Tiering

60

Remove

Glacier Flexible Retrieval (formerly...

1

Remove

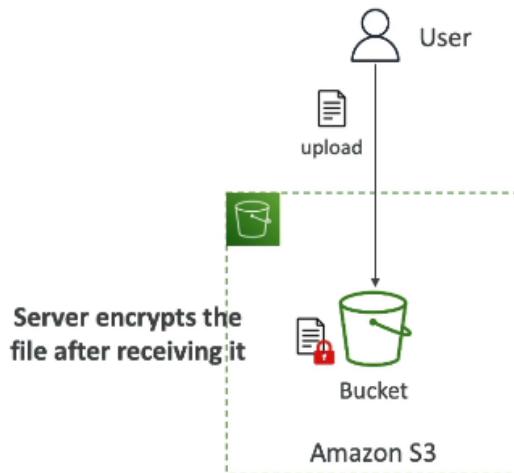
13.

## S3 Encryption

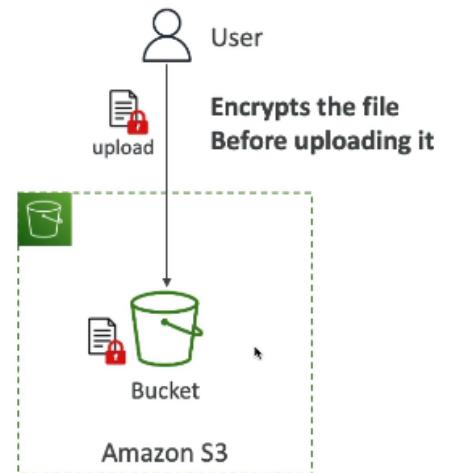
- Whenever we create a bucket ..by default it gets server side encryption

# S3 Encryption

## Server-Side Encryption (Default)



## Client-Side Encryption



2.

3. In here the both models exists but by default you should know that server-side encryption is always on.

## Shared responsibility Model for S3

### Shared Responsibility Model for S3



- Infrastructure (global security, durability, availability, sustainability, concurrent loss of data in two facilities)
- Configuration and vulnerability analysis
- Compliance validation
- S3 Versioning
- S3 Bucket Policies
- S3 Replication Setup
- Logging and Monitoring
- S3 Storage Classes
- Data encryption at rest and in transit

1.

## AWS Snow Family

# AWS Snow Family

- Highly-secure, portable devices to collect and process data at the edge, and migrate data into and out of AWS

- Data migration:



Snowcone



Snowball Edge



Snowmobile

- Edge computing:



Snowcone



Snowball Edge

- 1.
2. The two use cases are Data migration and Edge Computing
3. Bard definition of AWS snow

Sure. AWS Snow Family is a family of physical devices that transport data in and out of AWS. The devices are designed to be used in remote or challenging environments where there is limited or no network connectivity.

The AWS Snow Family includes three types of devices:

- **AWS Snowcone:** The smallest and most portable device in the Snow Family. It is a ruggedized, 4.5-pound device that can store up to 80TB of data.
  - **AWS Snowball:** A larger and more powerful device than Snowcone. It is available in two models: Compute Optimized and Storage Optimized. The Compute Optimized model has 80TB of storage and 16 vCPUs, while the Storage Optimized model has 100TB of storage and no vCPUs.
  - **AWS Snowmobile:** The largest and most powerful device in the Snow Family. It is a 45-foot long shipping container that can store up to 100PB of data.
- 4.
  5. Why do we need Data migration with aws snow..

6. If we look at the time takes to transfer over the network

	Time to Transfer		
	100 Mbps	1Gbps	10Gbps
10 TB	12 days	30 hours	3 hours
100 TB	124 days	12 days	30 hours
1 PB	3 years	124 days	12 days

7. Using over the network it can have many challenges ...like to transer 1 pb at 100mbps..it will take for ever

## Data Migrations with AWS Snow Family

### Challenges:

- Limited connectivity
- Limited bandwidth
- High network cost
- Shared bandwidth (can't maximize the line)
- Connection stability

- 8.

AWS Snow Family: offline devices to perform data migrations

If it takes more than a week to transfer over the network, use Snowball devices!

- 9.

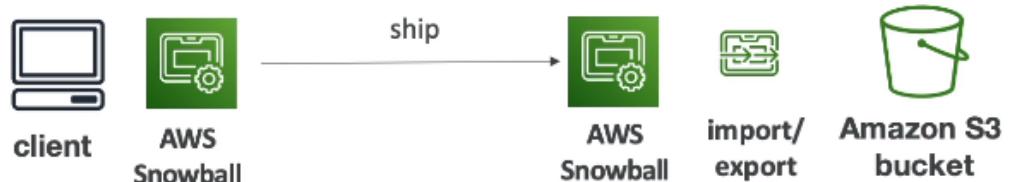
10. Aws will actually sends us the physical device via post..and we have to load data and send it back to aws
11. We can direct upload to amazon s3 bucket at 10Gbit/s..if our data is less
12. With snow daily..first client request aws snowball device...client receives it thru post...and client loads the data and ship it to aws..and in aws people will transfer the data into s3 bucket using their own infrastructure which is very fast

# Diagrams

- Direct upload to S3:



- With Snow Family:



13.

14. Transferring data physically is aws snowball route

15. Snowball edge

## Snowball Edge (for data transfers)



- Physical data transport solution: move TBs or PBs of data in or out of AWS
- Alternative to moving data over the network (and paying network fees)
- Pay per data transfer job
- Provide block storage and Amazon S3-compatible object storage
- **Snowball Edge Storage Optimized**
  - 80 TB of HDD capacity for block volume and S3 compatible object storage
- **Snowball Edge Compute Optimized**
  - 42 TB of HDD or 28TB NVMe capacity for block volume and S3 compatible object storage
- Use cases: large data cloud migrations, DC decommission, disaster recovery



16.

Block storage and S3 compatible object storage are two different types of storage that are available on AWS Snowball Edge devices.

**Block storage** is a type of storage that is designed for storing large files that need to be accessed quickly. Block storage is made up of a series of blocks, and each block can be accessed individually. This makes block storage ideal for applications that need to read and write large files frequently, such as databases and file servers.

**S3 compatible object storage** is a type of storage that is designed for storing large amounts of data that is accessed less frequently. S3 compatible object storage is made up of a series of objects, and each object can contain any type of data. This makes S3 compatible object storage ideal for applications that need to store large amounts of data such as media files and backup data.

AWS Snowball Edge devices offer both block storage and S3 compatible object storage. This allows you to choose the right type of storage for your needs. If you need to store large files that need to be accessed quickly, you can use block storage. If you need to store large amounts of data that is accessed less frequently, you can use S3 compatible

- 17.
18. Aws snowcone and snowcone ssd

## AWS Snowcone & Snowcone SSD



- Small, portable computing, anywhere, rugged & secure, withstands harsh environments
- Light (4.5 pounds, 2.1 kg)
- Device used for edge computing, storage, and data transfer
- Snowcone – 8 TB of HDD Storage
- Snowcone SSD – 14 TB of SSD Storage
- Use Snowcone where Snowball does not fit (space-constrained environment)
- Must provide your own battery / cables



19. • Can be sent back to AWS offline, or connect it to

Sure. AWS Snowcone and Snowcone SSD are two members of the AWS Snow Family of edge computing, edge storage, and data migration devices. They are designed to be used in remote or challenging environments where there is limited or no network connectivity.

**AWS Snowcone** is the smallest and most portable device in the Snow Family. It is a ruggedized, 4.5-pound device that can store up to 80TB of data. It has two vCPUs, 4 GB of memory, and a 1GbE network interface.

**AWS Snowcone SSD** is a newer version of Snowcone that has 14 TB of solid-state drive (SSD) storage. It has the same two vCPUs and 4 GB of memory as Snowcone, but it has a faster 10GbE network interface.

Both Snowcone and Snowcone SSD can be used to migrate data to or from AWS, to backup data, or to perform data analysis at the edge. The devices are easy to use and

20. ~~can be deployed quickly~~

21. AWS SnowMobile

## AWS Snowmobile



- Transfer exabytes of data (1 EB = 1,000 PB = 1,000,000 TBs)
- Each Snowmobile has 100 PB of capacity (use multiple in parallel)
- High security: temperature controlled, GPS, 24/7 video surveillance
- Better than Snowball if you transfer more than 10 PB

22. If we want to send 10EB(1000PB) we need 10 snowmobile trucks

# AWS Snow Family for Data Migrations



Snowcone



Snowball Edge



Snowmobile

	Snowcone & Snowcone SSD	Snowball Edge Storage Optimized	Snowmobile
Storage Capacity	8 TB HDD 14 TB SSD	80 TB usable	< 100 PB
Migration Size	Up to 24 TB, online and offline	Up to petabytes, offline	Up to exabytes, offline
DataSync agent	Pre-installed		
Storage Clustering		Up to 15 nodes	

23.

24. For snowball edge it has storage clustering ..to increase the snowball size to 15 nodes

## Snow Family – Usage Process

1. Request Snowball devices from the AWS console for delivery
2. Install the snowball client / AWS OpsHub on your servers
3. Connect the snowball to your servers and copy files using the client
4. Ship back the device when you're done (goes to the right AWS facility)
5. Data will be loaded into an S3 bucket
6. Snowball is completely wiped

25.

26. What is Edge computing?

27. We may have areas where it is hard to capture data..due to many challenges like no internet etc..due to moving truck,or a ship on the sea

28. To collect these data we use snowball edge

# What is Edge Computing?

- Process data while it's being created on an edge location
  - A truck on the road, a ship on the sea, a mining station underground...



- These locations may have
  - Limited / no internet access
  - Limited / no easy access to computing power
- We setup a Snowball Edge / Snowcone device to do edge computing
- Use cases of Edge Computing:
  - Preprocess data
  - Machine learning at the edge
  - Transcoding media streams
- Eventually (if need be) we can ship back the device to AWS (for transferring data for example)

29.

AWS Snowball Edge is a device that can be used to collect and process data at the edge. It is a ruggedized device that can be deployed in remote or disconnected locations. Snowball Edge has on-board storage and compute power, so it can be used to run applications and services locally. This can improve performance, reduce latency, and increase security.

Snowball Edge is available in two configurations:

30.

- **Snowball Edge Storage Optimized:** This configuration has 57.6 TB of storage and 16 vCPUs. It is designed for data transfer and offline processing.
- **Snowball Edge Compute Optimized:** This configuration has 7.68 TB of NVMe storage and 52 vCPUs. It is designed for local processing and edge computing workloads.

# Snow Family – Edge Computing

- Snowcone & Snowcone SSD (smaller)
  - 2 CPUs, 4 GB of memory, wired or wireless access
  - USB-C power using a cord or the optional battery
- Snowball Edge – Compute Optimized
  - 104 vCPUs, 416 GiB of RAM
  - Optional GPU (useful for video processing or machine learning)
  - 28 TB NVMe or 42TB HDD usable storage
- Snowball Edge – Storage Optimized
  - Up to 40 vCPUs, 80 GiB of RAM, 80 TB storage
  - Object storage clustering available
- All: Can run EC2 Instances & AWS Lambda functions (using AWS IoT Greengrass)
- Long-term deployment options: 1 and 3 years discounted pricing



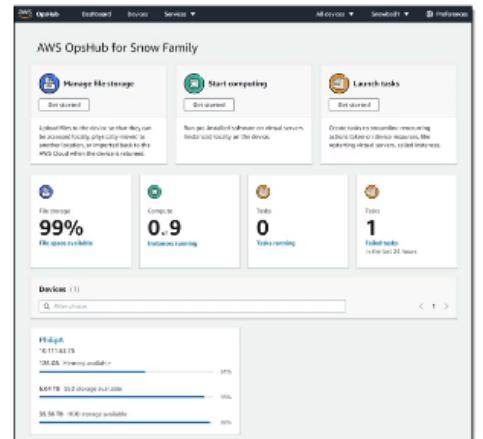
31.

32. every snowball edge devices can run ec2 instances

33. AWS OpsHub

## AWS OpsHub

- Historically, to use Snow Family devices, you needed a CLI (Command Line Interface tool)
- Today, you can use **AWS OpsHub** (a software you install on your computer / laptop) to manage your Snow Family Device
  - Unlocking and configuring single or clustered devices
  - Transferring files
  - Launching and managing instances running on Snow Family Devices
  - Monitor device metrics (storage capacity, active instances on your device)
  - Launch compatible AWS services on your devices (ex: Amazon EC2 instances, AWS DataSync, Network File System (NFS))



<https://aws.amazon.com/blogs/aws/aws-snowball-edge-updates/>

## AWS Snow Family Handson

1. Refer online

## Storage Gateway Overview

1. In AWS we have hybrid cloud for storage
2. Where part of our infrastructure will be on premises and part of your infrastructure will be on cloud
3. S3 is a proprietary(meaning : relating to owner or owners) storage technology

## Hybrid Cloud for Storage

- AWS is pushing for "hybrid cloud"
  - Part of your infrastructure is on-premises
  - Part of your infrastructure is on the cloud
- This can be due to
  - Long cloud migrations
  - Security requirements
  - Compliance requirements
  - IT strategy
- S3 is a proprietary storage technology (unlike EFS / NFS), so how do you expose the S3 data on-premise?
- AWS Storage Gateway!

4.

## AWS Storage Cloud Native Options



5.

# AWS Storage Gateway

- Bridge between on-premise data and cloud data in S3
- Hybrid storage service to allow on-premises to seamlessly use the AWS Cloud
- Use cases: disaster recovery, backup & restore, tiered storage
- Types of Storage Gateway:
  - File Gateway
  - Volume Gateway
  - Tape Gateway
- No need to know the types at the exam

6.

7. Also learn more about aws storage gateway in chatgpt or bard



## Amazon S3 summary

### Amazon S3 – Summary

- Buckets vs Objects: global unique name, tied to a region
- S3 security: IAM policy, S3 Bucket Policy (public access), S3 Encryption
- S3 Websites: host a static website on Amazon S3
- S3 Versioning: multiple versions for files, prevent accidental deletes
- S3 Replication: same-region or cross-region, must enable versioning
- S3 Storage Classes: Standard, IA, 1Z-IA, Intelligent, Glacier (Instant, Flexible, Deep)
- Snow Family: import data onto S3 through a physical device, edge computing
- OpsHub: desktop application to manage Snow Family devices
- Storage Gateway: hybrid solution to extend on-premises storage to S3

1.

## Extras

1. Should learn more about snow family