S3 - Simple Storage Service

- Infinitely scaling storage
- A lot of other AWS services use S3 for storage integration and many websites rely on S3.
- Use cases
  - Backup and storage
  - Disaster Recovery
  - Data archival (cost optimization)
  - Hybrid cloud storage (switching from on-premises storage to the cloud)
  - Host applications, media
  - Big data analytics
  - To host static websites
- S3 allows you to store objects (files) in buckets (directories or folders)
- A bucket needs to have a globally unique name. S3 is in fact the only service which requires you to have a globally unique name.
- Buckets can be defined at the region level. (you might end up thinking its a global service due to the unique naming but its regional).

S3 Objects

- Anything within a S3 bucket is an object. These objects have a key.
- Each object in S3 is uniquely identified within a bucket by a key. The key is essentially the name of the object within the bucket.
- **The key is the FULL path. It is composed of prefix + object name**

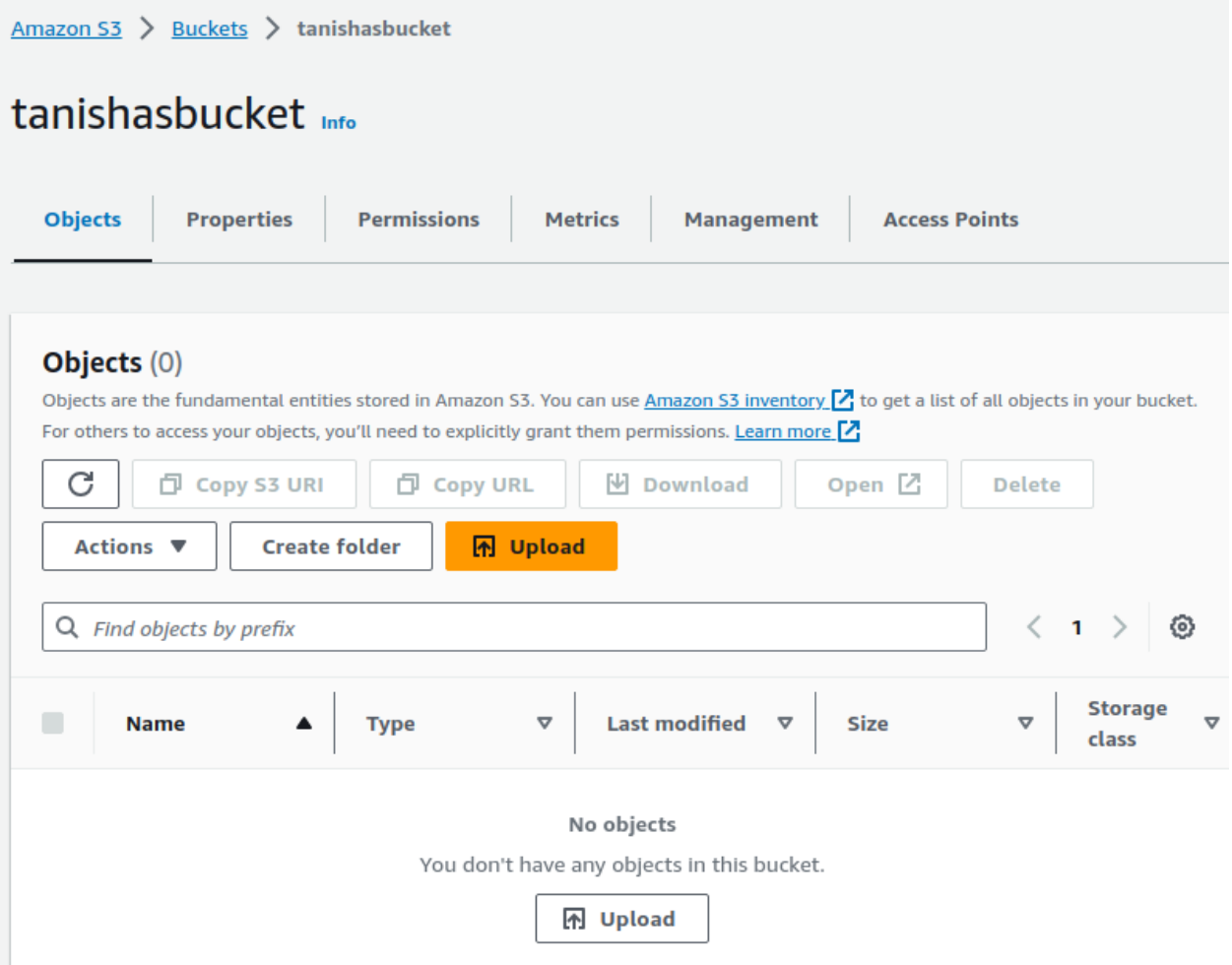
    - ex: s3://my-bucket/folder_a/folder_b/my_file.txt
    - here, you can decode this path (key) and figure out that the object we are referring to through this key is `my_file.txt`
- There is no concept of directories or folders in S3, it's all keys and objects. The UI tricks you to think otherwise though as it lets you create directory like structure (a folder structure).
- Don't get confused

- **Bucket:** A container for objects stored in Amazon S3. It's like a top-level folder.
- **Object:** The fundamental entity stored in Amazon S3. It consists of data, a key (unique within a bucket), and metadata.
- **Key:** The unique identifier for an object within a bucket. It is essentially the name of the object.

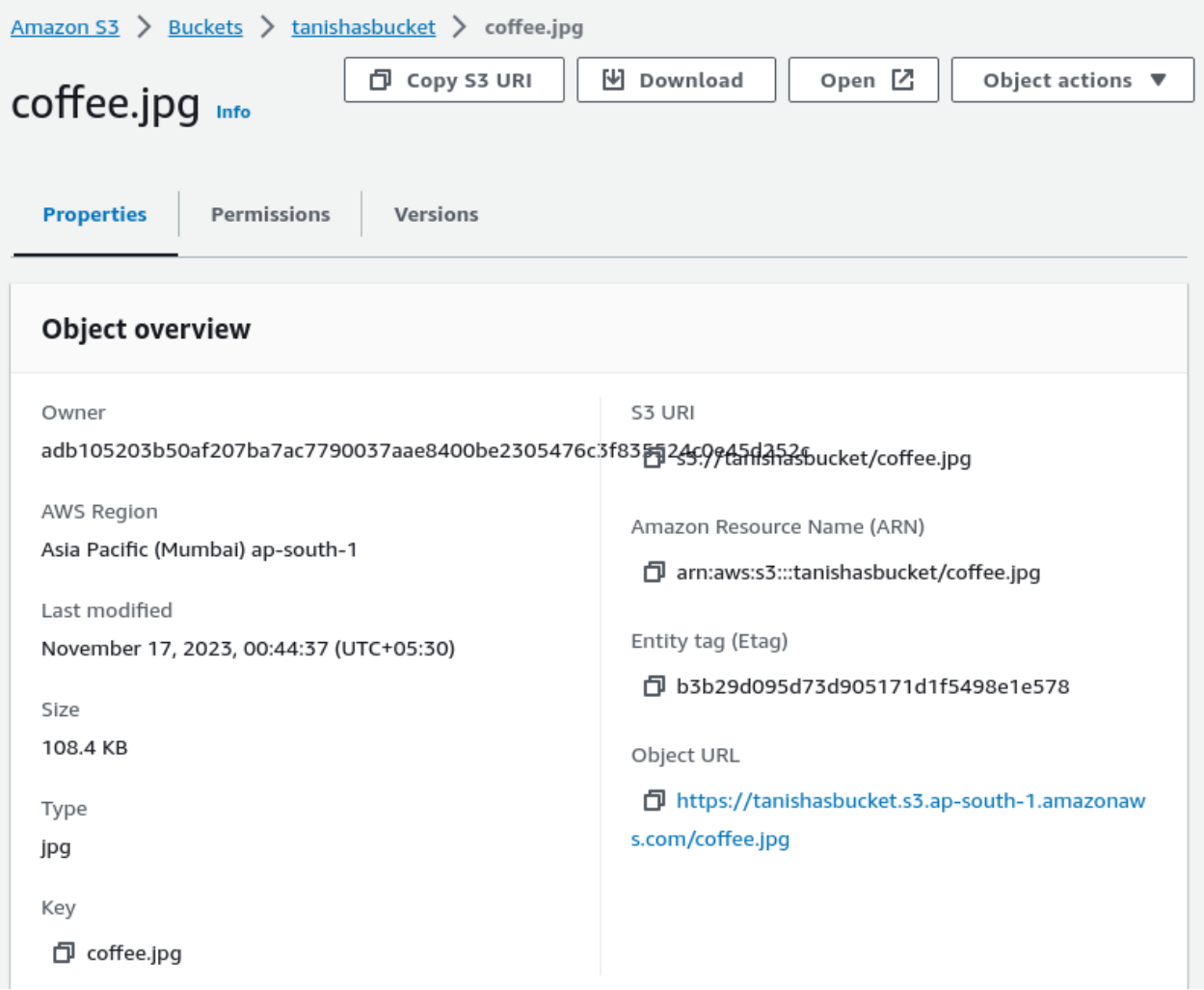
- Max object size is 5TB, if you need to upload data larger than 5GB you need to use multi-part upload.
- Metadata stored w each object is info about the data stored or user-defined info like author details, description, etc.

S3 basic Hands-on

- keep the settings as default, name the bucket and simply create it.
- upload to add files



- Once uploaded, try accessing the file using the public object URL. Access denied.



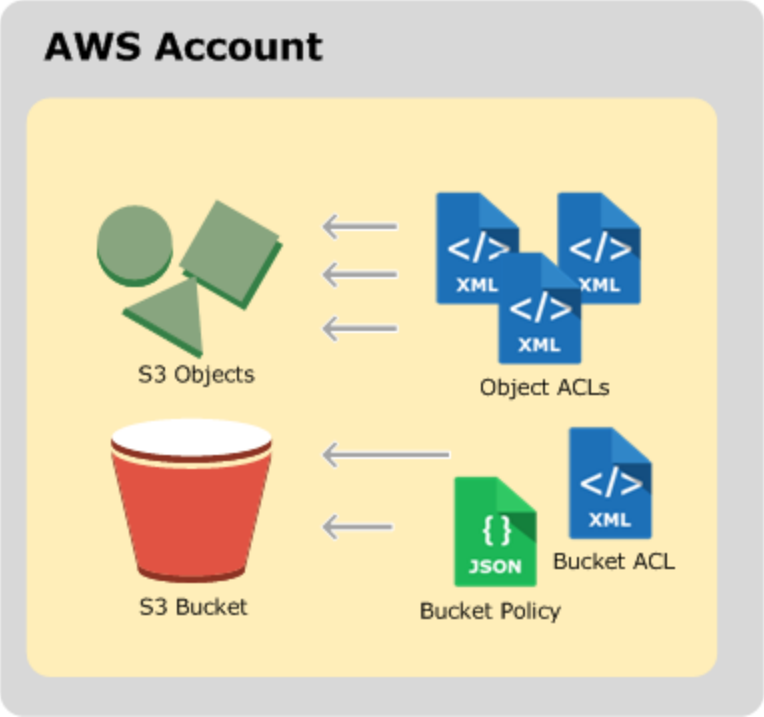
- We can make an object public to let others access it. Once we allow public access, we can create a pre-signed URL for a certain amount of time (x minutes or hours or days) and share it.

S3 Security

- S3 buckets can be secured in several ways.
    - **User based - IAM Policies** - defines which API calls should be allowed for a specific user in a group - an IAM principal can access an S3 object only

if the user's permissions are set to `allow` or the resource policy allows it AND there is no explicit `deny`.

- Object encryption using encryption keys.
- Resource based
    1. **Bucket Policies (IMP)** - bucket wide rules which can be applied to the bucket as a whole and also the objects within the bucket. It can be defined using the S3 console, these policies allow cross account usage.
    2. Bucket ACL (Access control list) - rules of access for a bucket as a whole (less common)
    3. Object ACL - rules of access at the object level



- **S3 BUCKET POLICIES**

  "*" = all

  It is a JSON based policy. The policy consists of a few key-pair values like

- 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

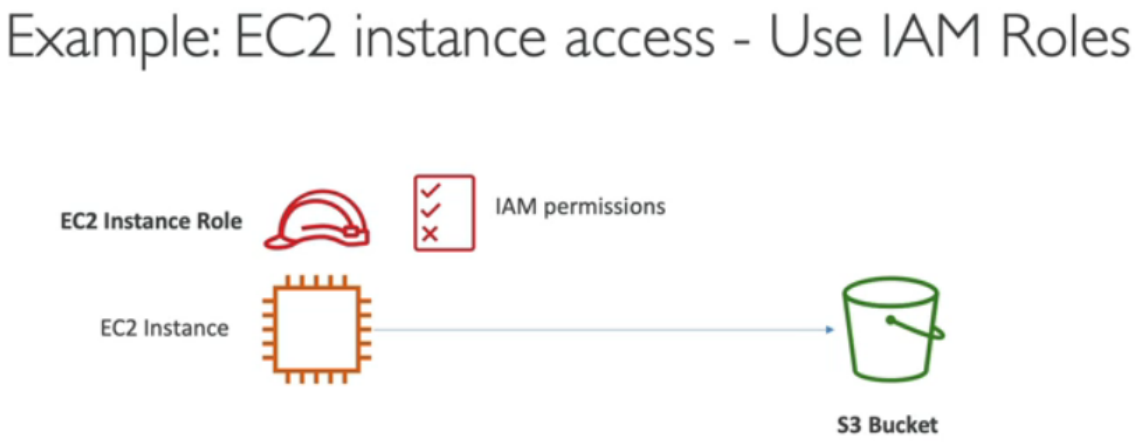
```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicRead",
            "Effect": "Allow",
            "Principal": "*",
            "Action": [
                "s3:GetObject"
            ],
            "Resource": [
                "arn:aws:s3:::examplebucket/*"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

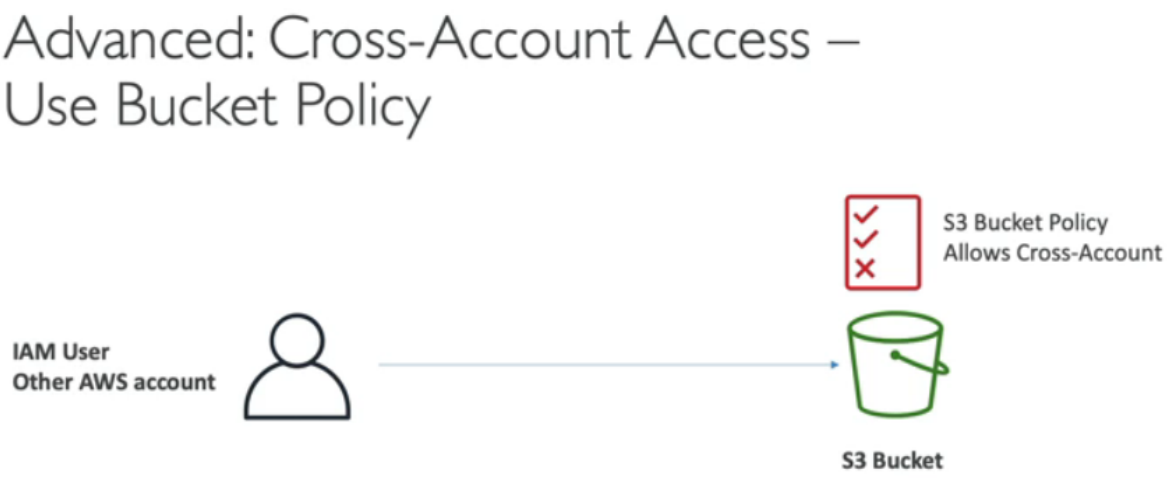
In the policy above, anyone can access the objects within the examplebucket and use the GetObject method.

- Some Bucket Security examples:

Example: Public Access - Use Bucket Policy

Example: User Access to S3 – IAM permissions



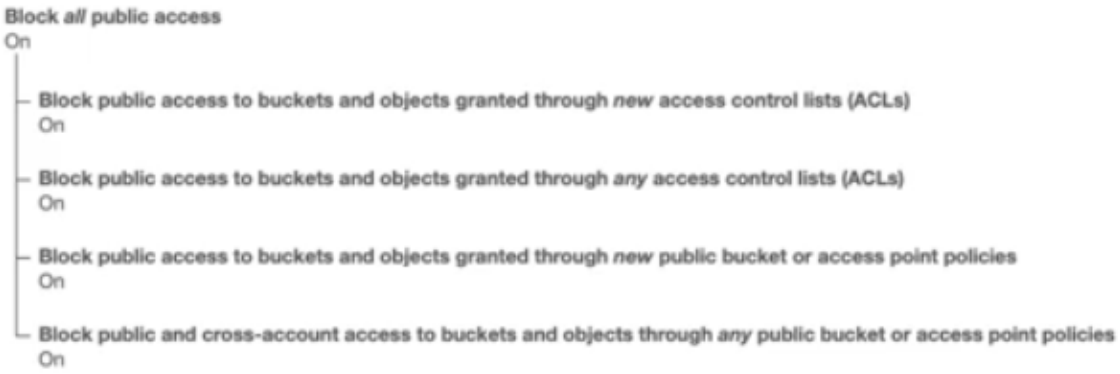


- **Bucket settings for “Block Public Access”**

  These settings were implemented by aws as an extra layer of security to prevent company data leaks.

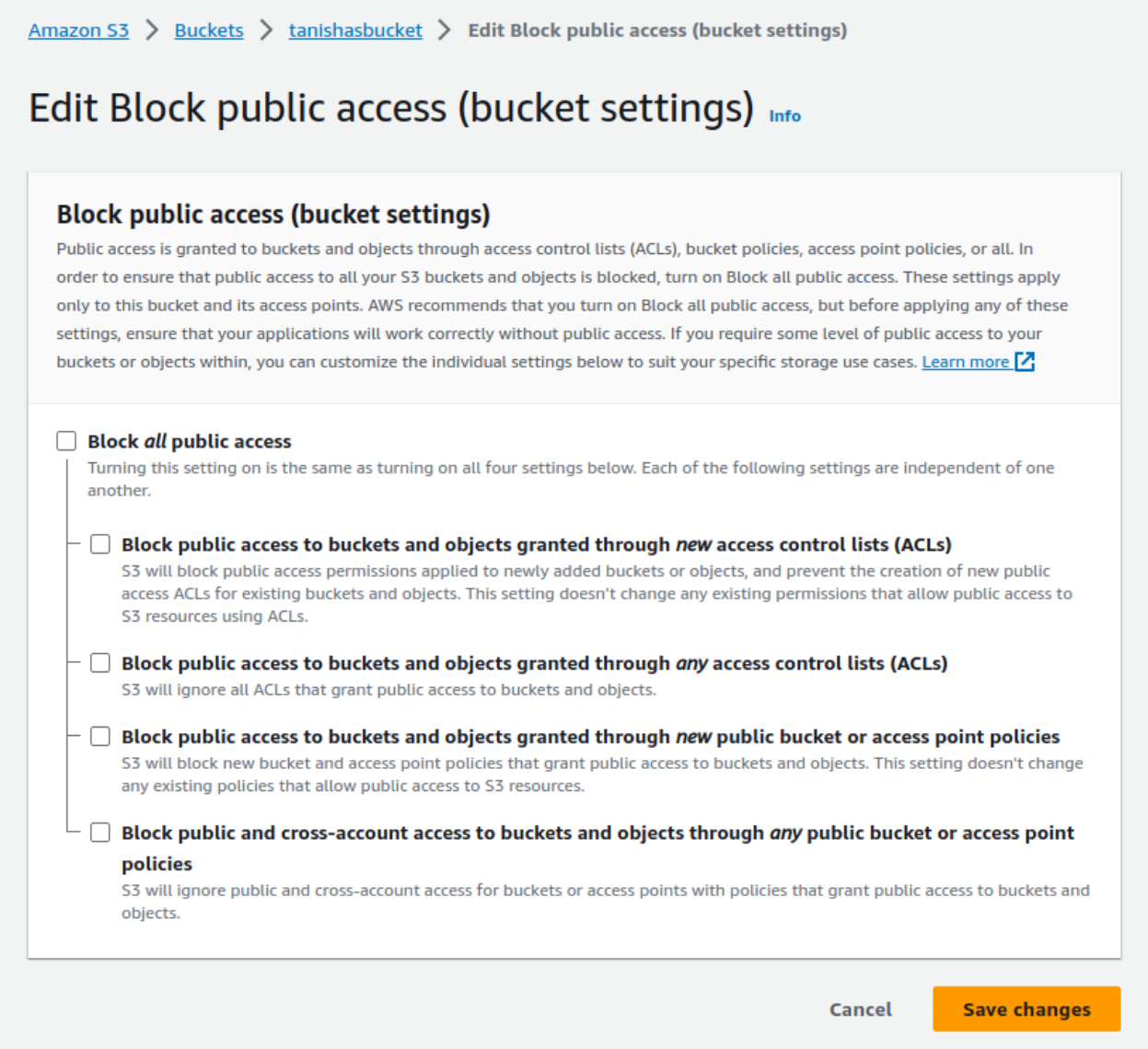
  Now, if you enable these settings and create a S3 bucket policy that makes it public, even then, the bucket will never be public.

  You can also implement these settings at the account level if you want none of your buckets to be public ever.

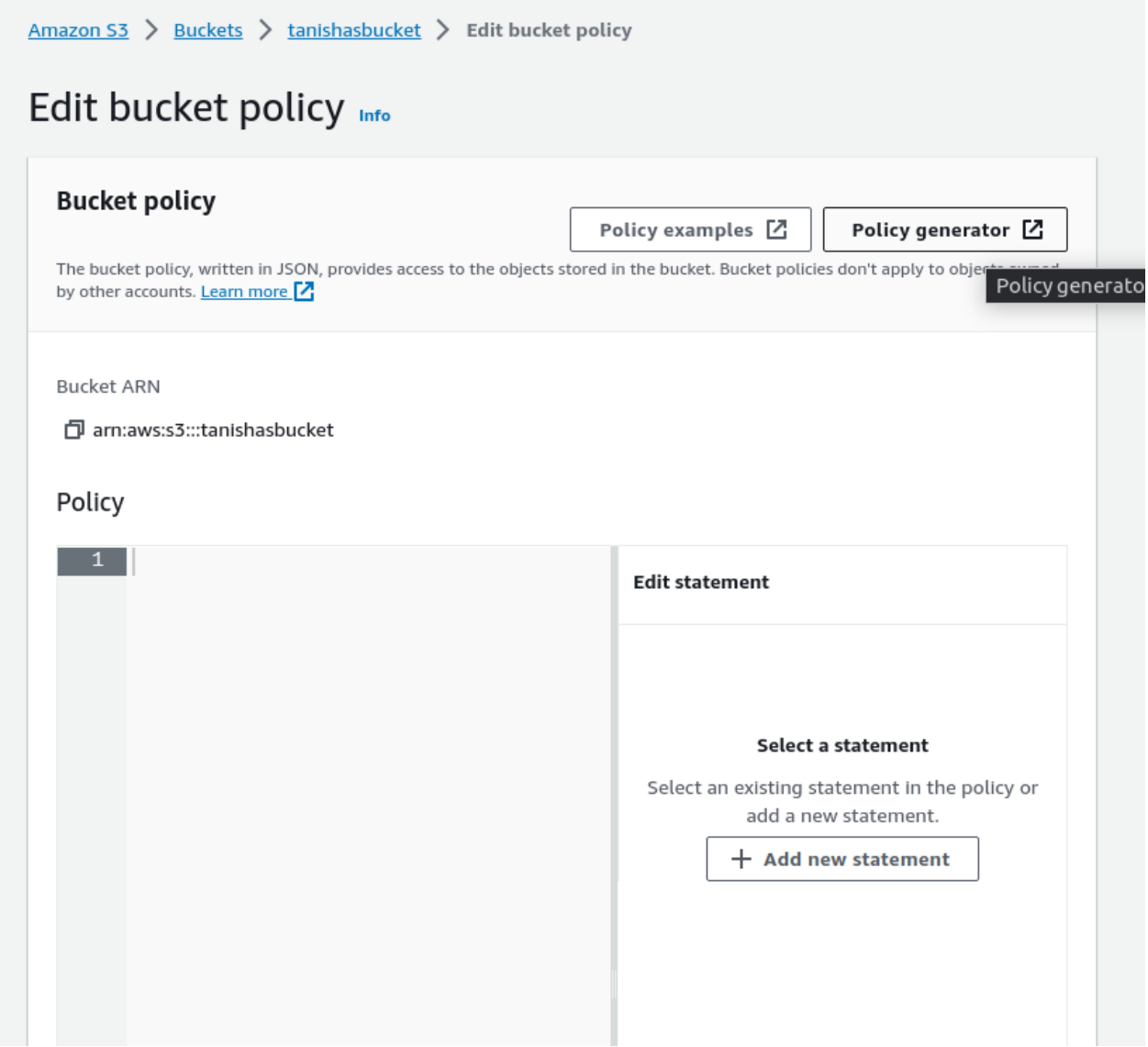


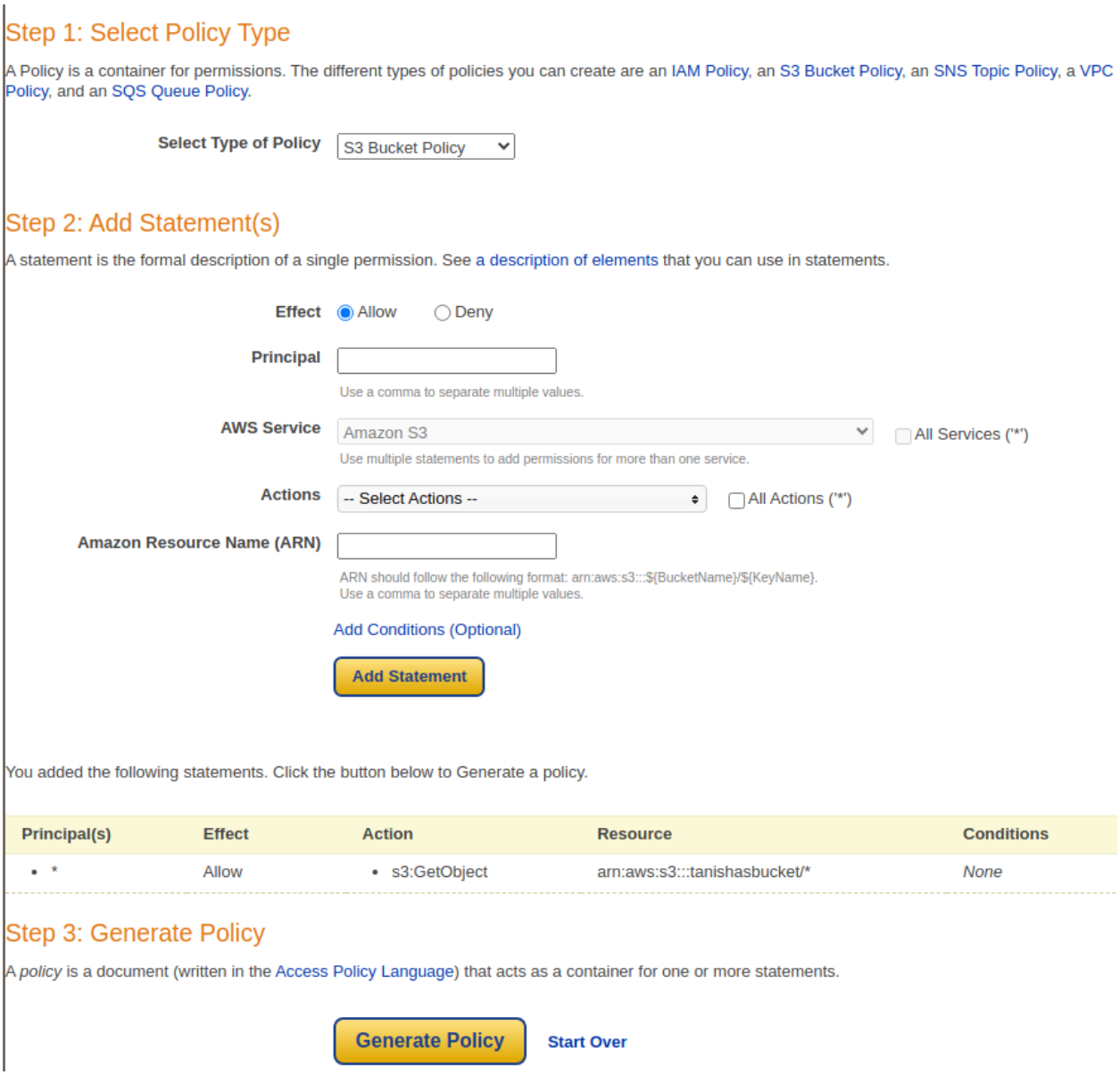
S3 Security Hands-on

- Let’s create a s3 bucket policy which lets anyone access the coffee.jpg image using the public url.
- bucket >> permissions tab
- Edit Block public access (bucket settings) and untick the checkbox to allow public access.



- Edit the bucket policy using the policy generator


Amazon S3 > Buckets > tanishasbucket > Edit bucket policy
Edit bucket policy Info
Bucket policy
Policy examples
Policy generator
The bucket policy, written in JSON, provides access to the objects stored in the bucket. Bucket policies don't apply to objects owned by other accounts. Learn more
Policy generato
Bucket ARN
arn:aws:s3:::tanishasbucket
Policy
1
Edit statement
Select a statement
Select an existing statement in the policy or add a new statement.
+ Add new statement



- Set the ARN equal to your bucket ARN + “/*” to allow access to all objects within the bucket.
- Copy and paste the generated JSON code of the policy into the console

Bucket ARN

arn:aws:s3:::tanishasbucket

Policy

```
{
  "Id": "Policy1700223159900",
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "Stmt1700223086616",
      "Action": [
        "s3:GetObject"
      ],
      "Effect": "Allow",
      "Resource": "arn:aws:s3:::tanishasbucket/
      "Principal": "*"
    }
  ]
}
```

Edit statement

Select a statement

Select an existing statement in the policy or add a new statement.

+ Add new statement

Amazon S3 > Buckets > tanishasbucket

tanishasbucket Info

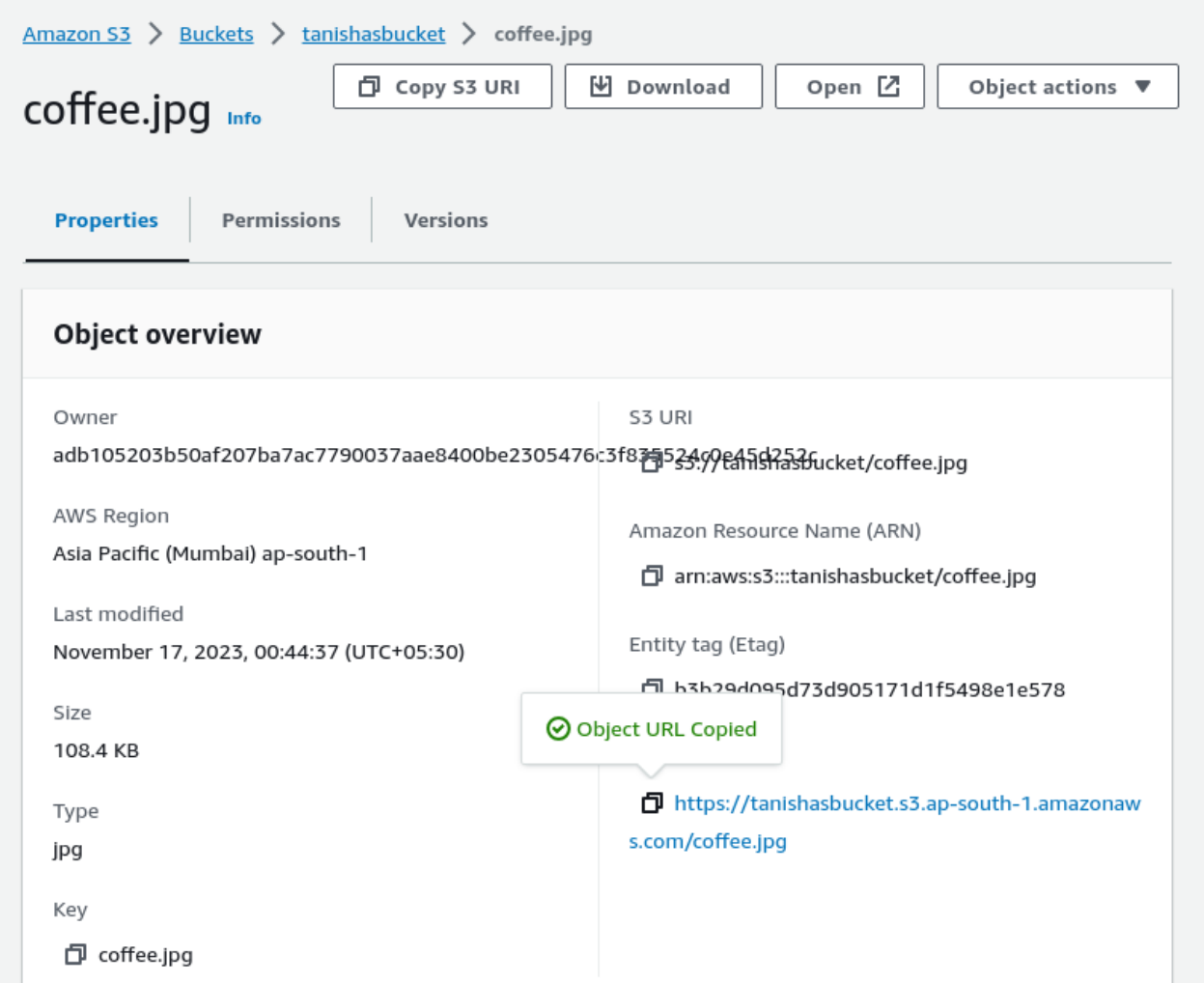
Publicly accessible

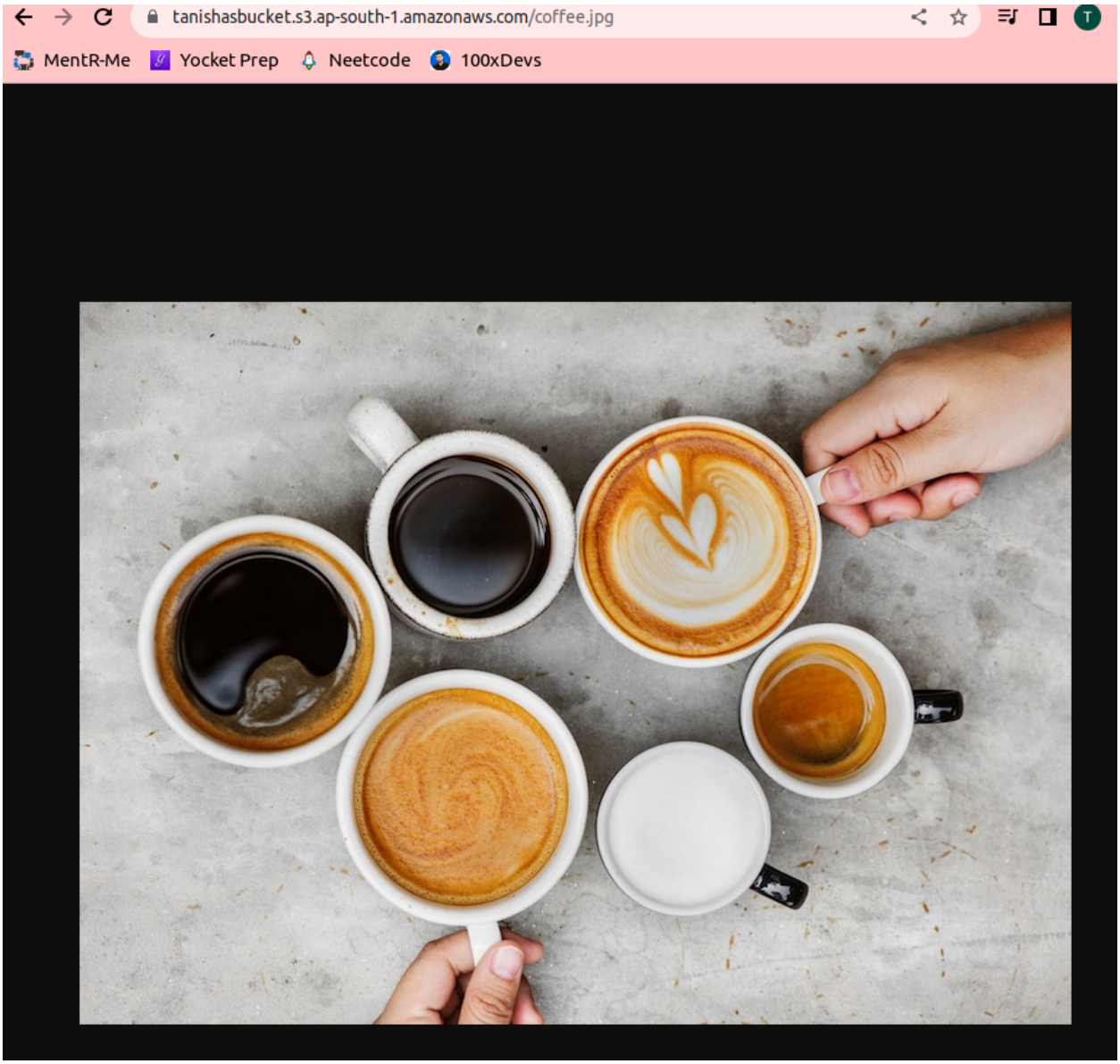
Objects | Properties | Permissions | Metrics | Management | Access Points

Permissions overview

Access

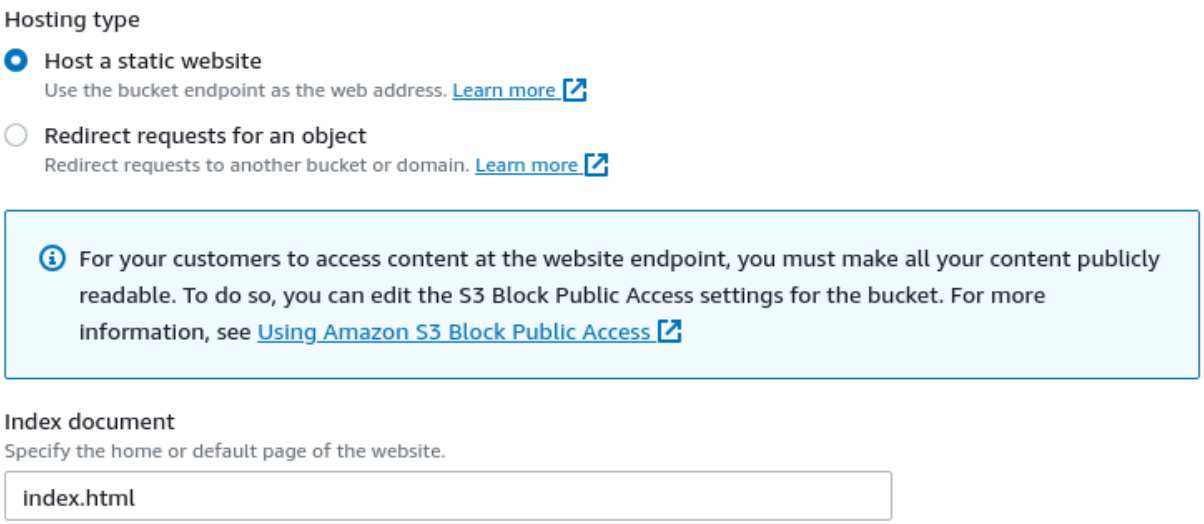
⚠ Public


Amazon S3 > Buckets > tanishasbucket > coffee.jpg
coffee.jpg Info
Copy S3 URI
Download
Open
Object actions
Properties
Permissions
Versions
Object overview
Owner
adb105203b50af207ba7ac7790037aae8400be2305476c3f8355524c0e45d252c
AWS Region
Asia Pacific (Mumbai) ap-south-1
Last modified
November 17, 2023, 00:44:37 (UTC+05:30)
Size
108.4 KB
Type
jpg
Key
coffee.jpg
S3 URI
s3://tanishasbucket/coffee.jpg
Amazon Resource Name (ARN)
arn:aws:s3:::tanishasbucket/coffee.jpg
Entity tag (Etag)
Object URL Copied
https://tanishasbucket.s3.ap-south-1.amazonaws.com/coffee.jpg

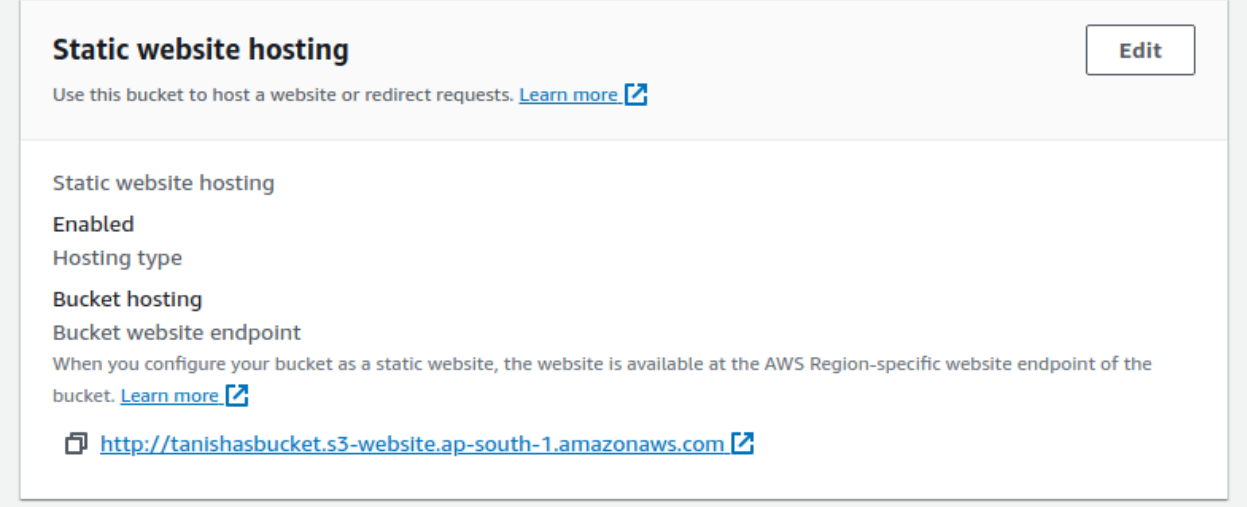


S3 Static Website Hosting

- S3 can host static websites and make them accessible on the internet.
- This will only be possible if you allow public access or public reads.
- 403 forbidden error == bucket isn't public
- Bucket >> properties >> static website hosting > edit > enable



- Add an index.html file to your bucket because obviously.
- Once, you have uploaded it, go back to the properties tab and then look for the link generated within static website hosting.

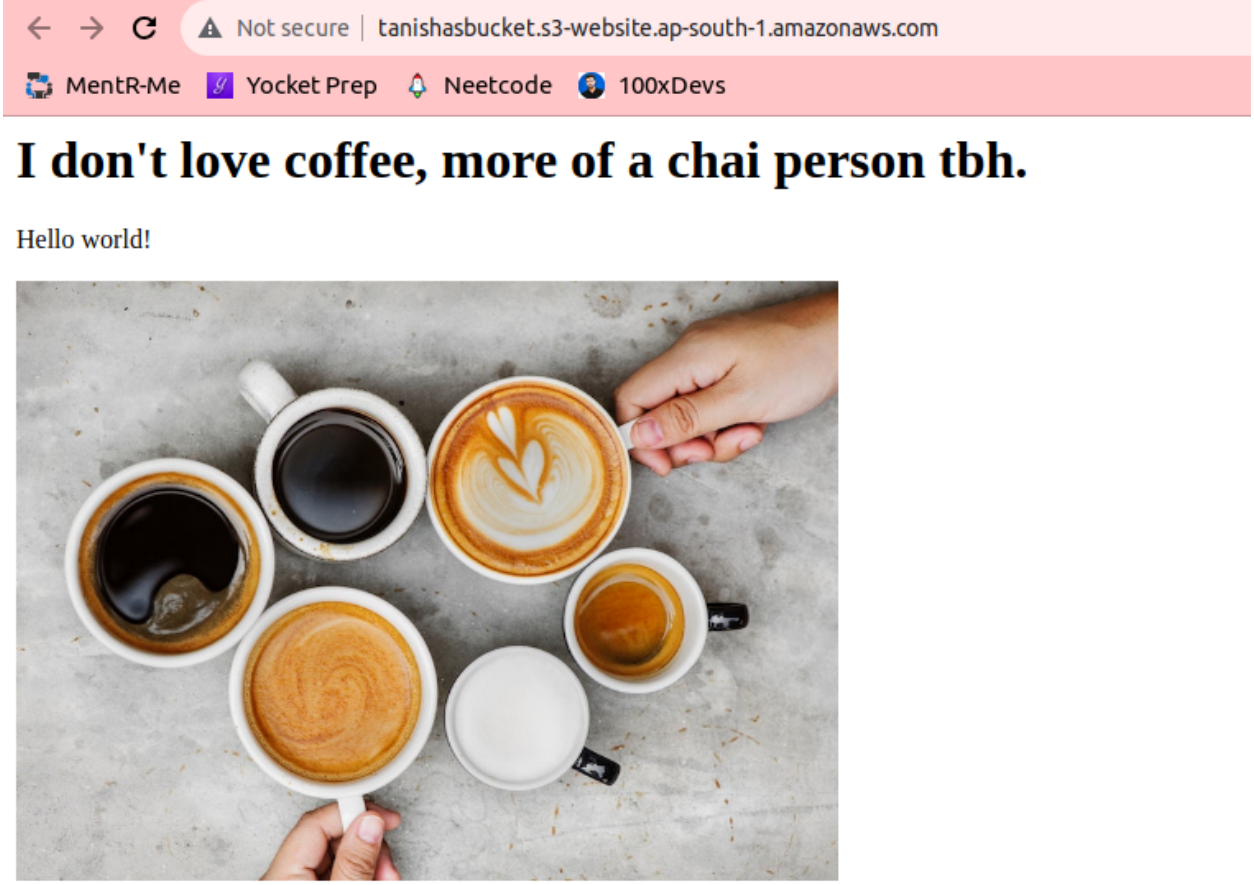


S3 Versioning

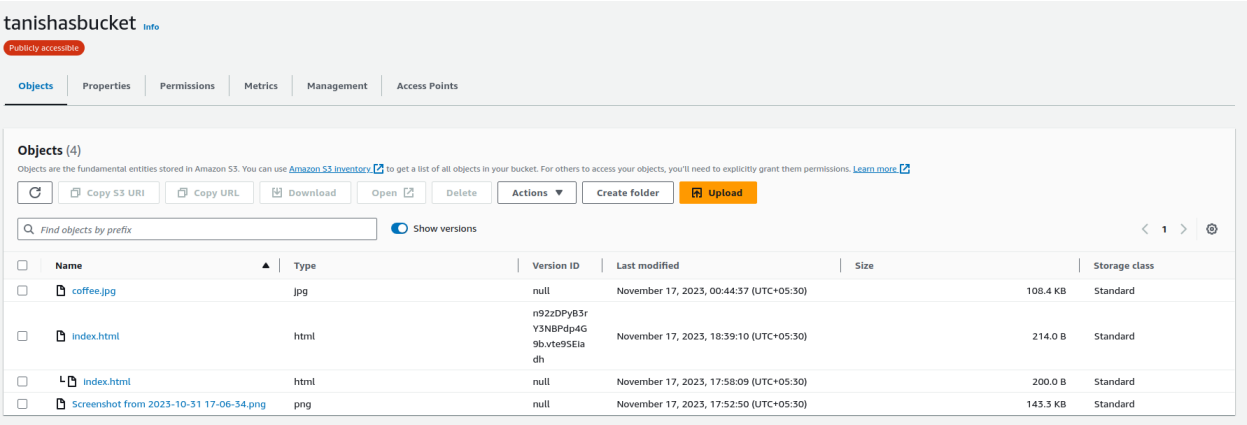
- You can version your files in S3. The versioning setting can be enabled at the bucket level.

- Once you enable it, a version of your object is created at that point in time. As and when you make any changes to that object, instead of overwriting the older version, it will create more versions.
- This is basically GIT yaar.
- Versioning is recommended because
    - you can restore accidentally deleted files
    - you can roll back to previous versions anytime you want.
- If you suspend versioning, the previous versions will still be available.

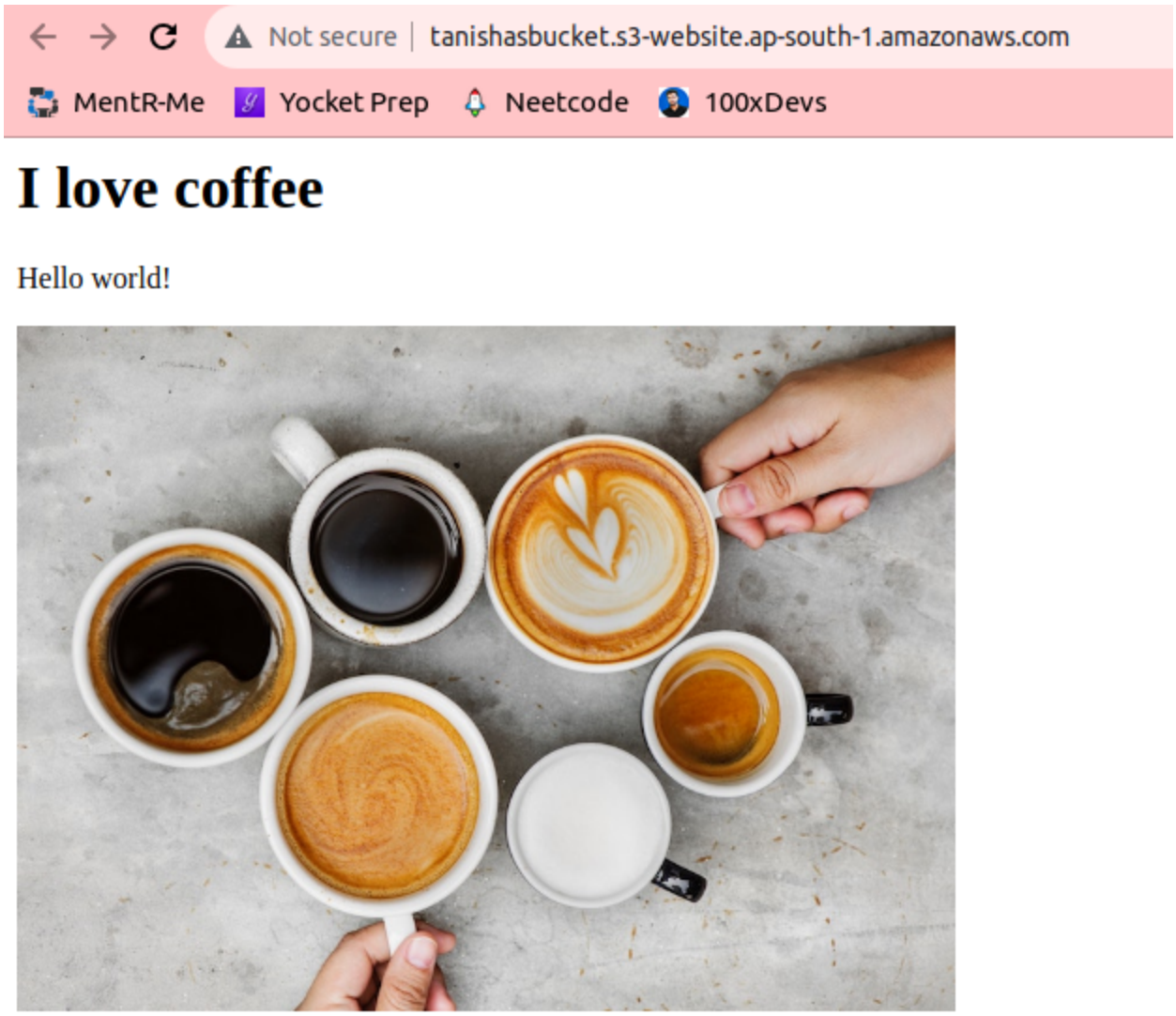
- Bucket >> edit bucket versioning >> enable >> save
- Any changes you make now will be stored in newer versions instead of being overwritten.
- Let's make changes to the index.html file locally and then upload the updated file to the bucket.
- If you visit the website again, it might seem like the code has been overwritten



- BUT at the backend, you can notice two versions of the index.html file. One being null (before versioning was enabled) and another with an alphanumeric ID.



- Now, if you wanna roll back to a previous version, delete the versions after that and refresh your webpage.

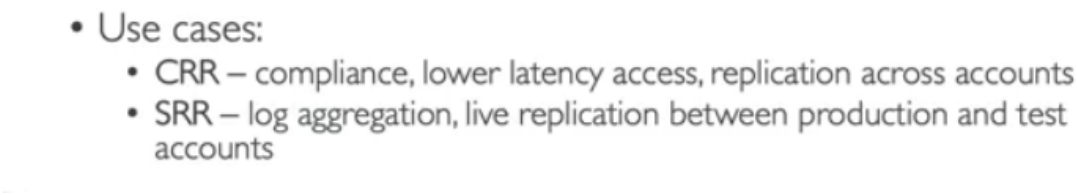


- Stephane did something, he deleted the coffee.jpg using a delete marker which somehow did something and the image was still accessible by enabling the show versions setting. Nahi samjha. Not that imp anyway. He deleted the delete marker to restore the image. kya ho raha hai bhai.

S3 Replication (CRR & SRR)

- CRR = cross region replication and SRR = same region replication

- The idea is that we have a source bucket in one region and a target bucket in another region and we want to implement asynchronous replication between those buckets.
- Enable versioning for both the buckets for this to work.
- Also, both these buckets can be in different AWS accounts and you obviously have to give proper read and write IAM permissions to both the S3 buckets.

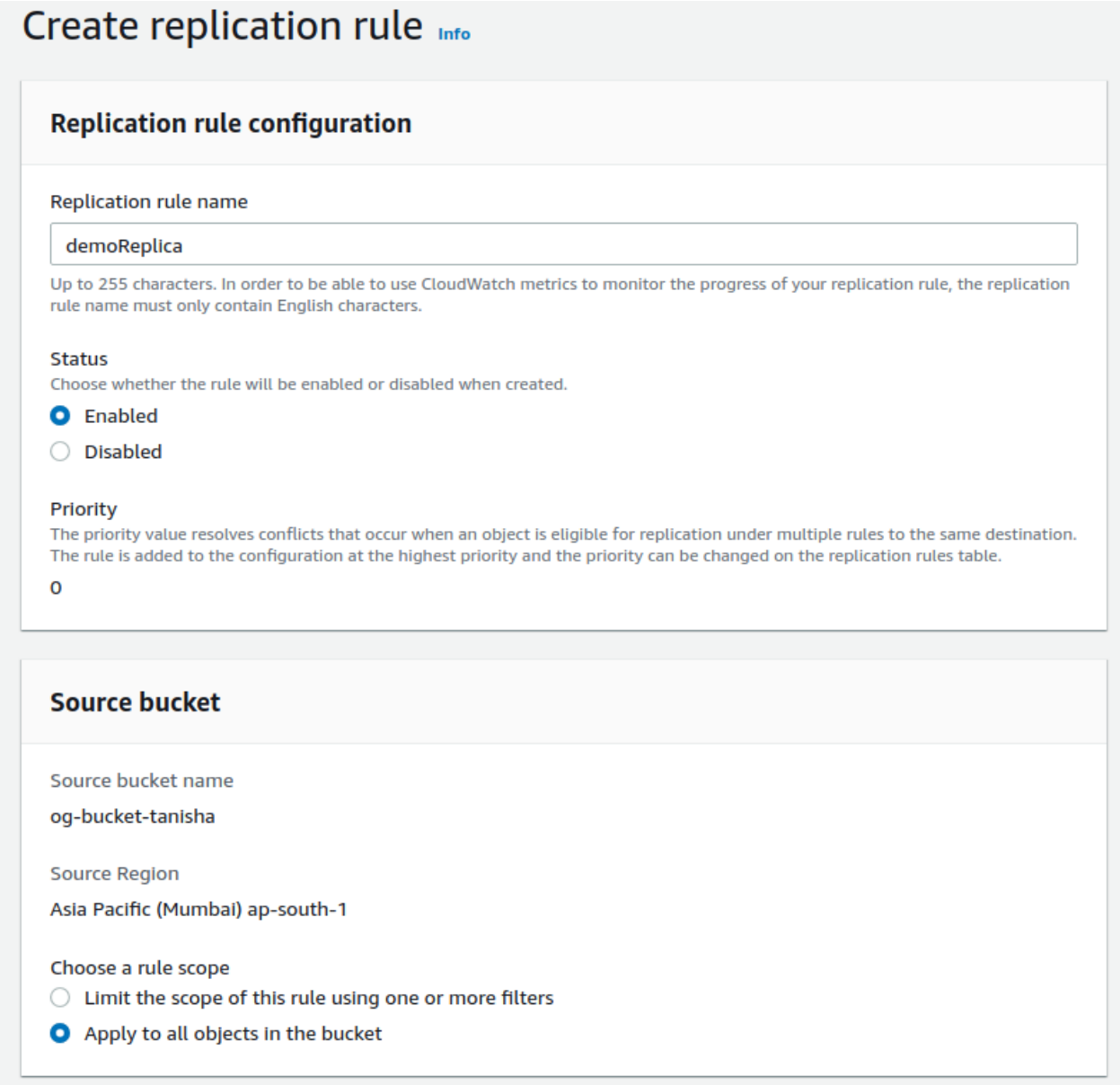


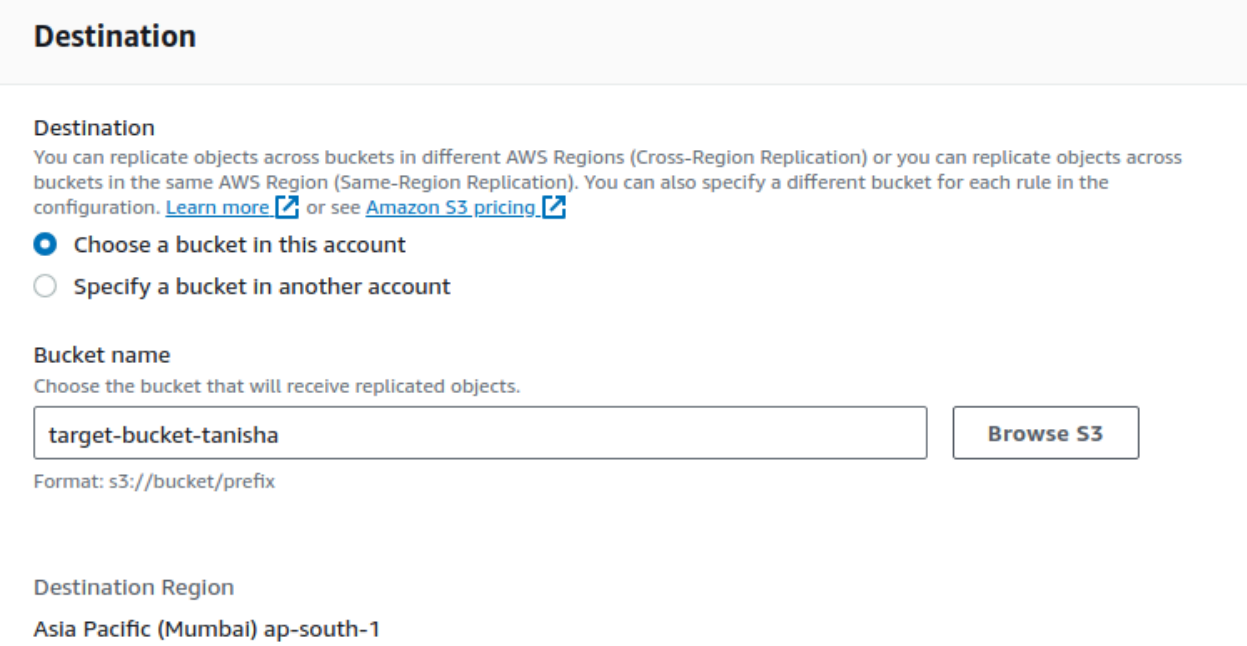
- **S3 Replication Hands-on**
- Create 2 buckets - OG bucket and target bucket.
- Enable versioning for both.
- You can keep both the buckets in the same region or diff regions.

○	og-bucket-tanisha	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	November 17, 2023, 19:05:50 (UTC+05:30)
○	tanishasbucket	Asia Pacific (Mumbai) ap-south-1	⚠ Public	November 17, 2023, 00:39:34 (UTC+05:30)
○	target-bucket-tanisha	Asia Pacific (Mumbai) ap-south-1	Bucket and objects not public	November 17, 2023, 19:06:10 (UTC+05:30)

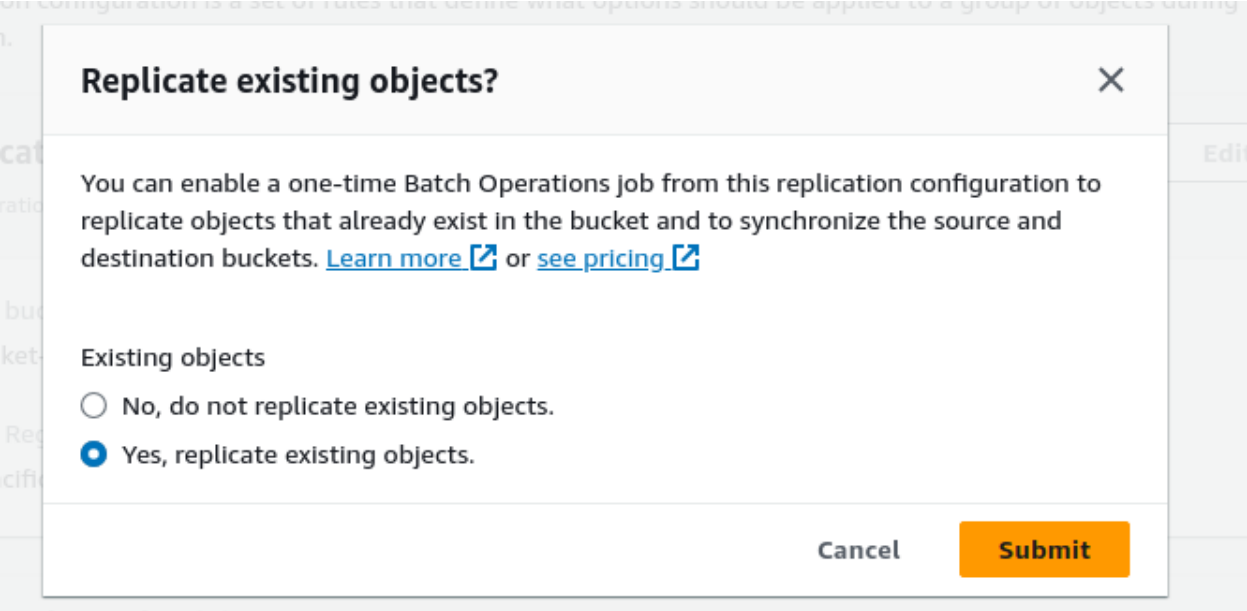
- Add a file to the og bucket
- Now let's set up replication

- og bucket >> management >> create replication rule



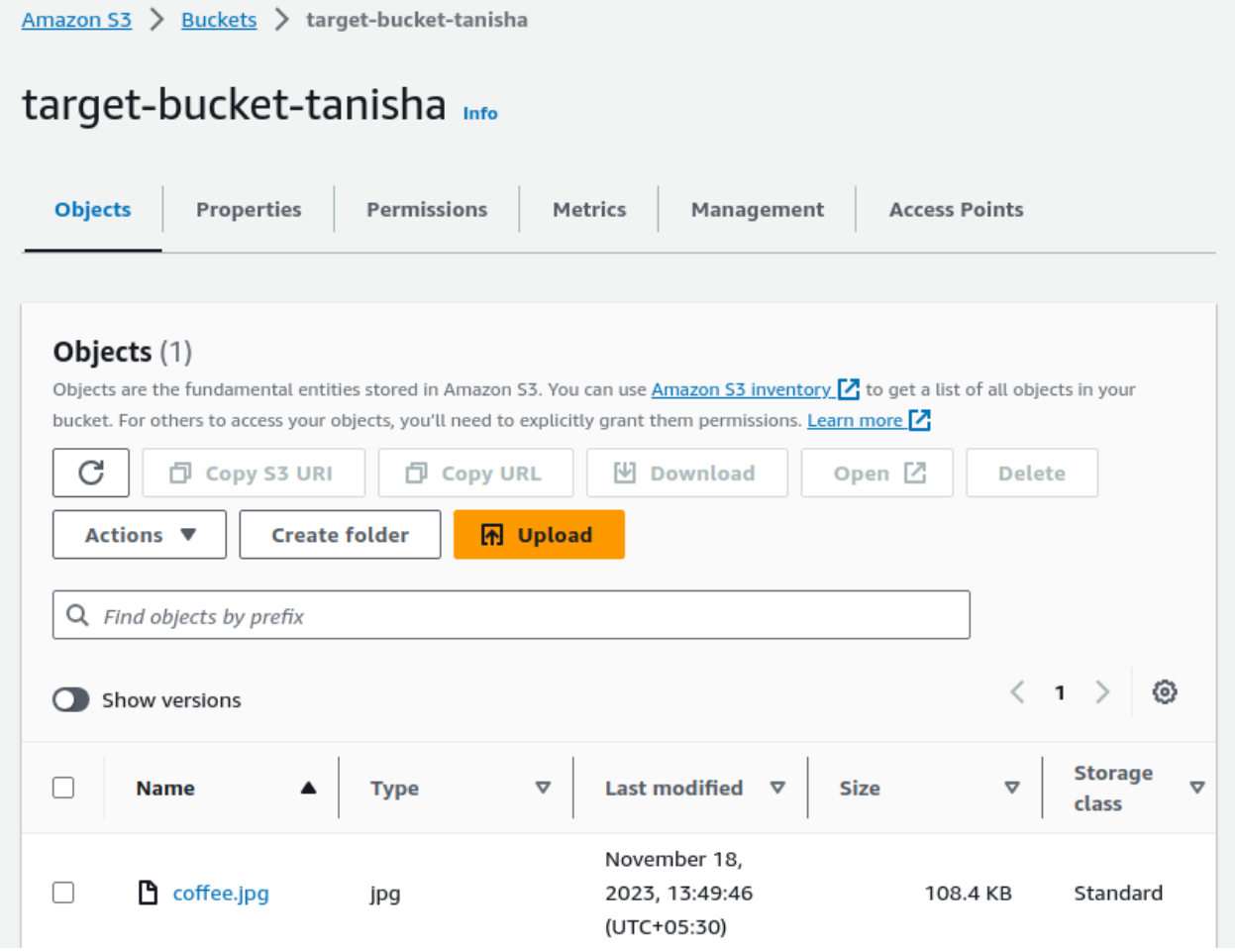


- Create a new IAM role for this.
- Once you set up replication rules, it replicates all the objects added to the source bucket from that point onwards. If you want it to replicate the already existing objects, select so in this prompt



- You are supposed to select no for now.

- If you check your target bucket, you will notice, no objects have been replicated yet. This is because we selected no.
- Upload new objects to the source bucket and then check the target bucket again.
- Coffee.jgp has been replicated. yay.



- Even the version IDs for an object will be the same in both the buckets. This could be super useful in cases of disaster recovery.

S3 Storage Classes

- S3 provides different types of storage classes for different purposes. This allows cost optimization and resource utilization at best.
- We can move our objects or buckets between these classes manually or by setting up S3 Lifecycle Rules / configurations to move objects automatically

- **Durability =** Durability refers to the ability of a storage system to protect data from loss or corruption over time. In the context of S3, it is expressed as a percentage and represents the likelihood that an object stored in S3 will not be lost over a given year.

  All S3 storage classes provide high durability, around **99.999999999% (11 nines)**. This means that the chance of losing an object stored in S3 due to system failures, hardware errors, or other issues is extremely low.
- **Availability** = Availability measures the accessibility of stored data. It indicates the percentage of time that the storage service is operational and can serve requests.

  Different S3 storage classes have varying levels of availability.

  All of the S3 storage classes provide around the same availability (99.99%) with varying retrieval times EXCEPT the one-zone IA storage class as the data is stored at a single AZ so availability drops to 99.5%.

1. **S3 Standard**

- 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…

2. **S3 Infrequent Access (IA)**

- For data that is less frequently accessed, but requires rapid access when needed
- Lower cost than S3 Standard

    1. Standard IA
        a. 99.99% available
        b. Used for disaster recovery and backups
    2. One Zone IA
        a. Data stored in a single AZ, so, it is lost when AZ is destroyed.
        b. Used for data that can be easily recreated or for secondary backup of on-premises data.
3. **S3 Glacier**
    - Low cost storage solution for archival or long term backup
    - It is priced for storage + the object retrieval cost
    - It has 3 options:

- 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

4. **S3 Intelligent Tiering**
   - Intelligent Tiering basically allows you to set up rules associated with each storage class and then it monitors and observes the data and then **auto-tiers** it to suitable storage classes.
   - There are no retrieval chargers for this.
   - For example:

- *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

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

S3 Storage Classes Hands-on

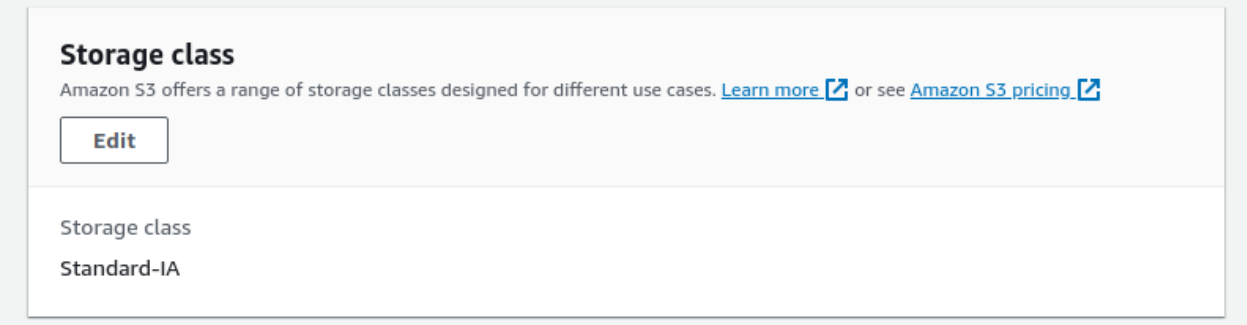
- Create a bucket
- Upload an object
- Before confirming the upload, click on the properties section and have a look at the storage classes.

	Storage class	Designed for	Availability Zones	Min storage duration
◉	Standard	Frequently accessed data (more than once a month) with milliseconds access	≥ 3	-
○	Intelligent-Tiering	Data with changing or unknown access patterns	≥ 3	-
○	Standard-IA	Infrequently accessed data (once a month) with milliseconds access	≥ 3	30 days
○	One Zone-IA	Recreatable, infrequently accessed data (once a month) stored in a single Availability Zone with milliseconds access	1	30 days
○	Glacier Instant Retrieval	Long-lived archive data accessed once a quarter with instant retrieval in milliseconds	≥ 3	90 days
○	Glacier Flexible Retrieval (formerly Glacier)	Long-lived archive data accessed once a year with retrieval of minutes to hours	≥ 3	90 days
○	Glacier Deep Archive	Long-lived archive data accessed less than once a year with retrieval of hours	≥ 3	180 days
○	Reduced redundancy	Noncritical, frequently accessed data with milliseconds access (not recommended as S3 Standard is more cost effective)	≥ 3	-

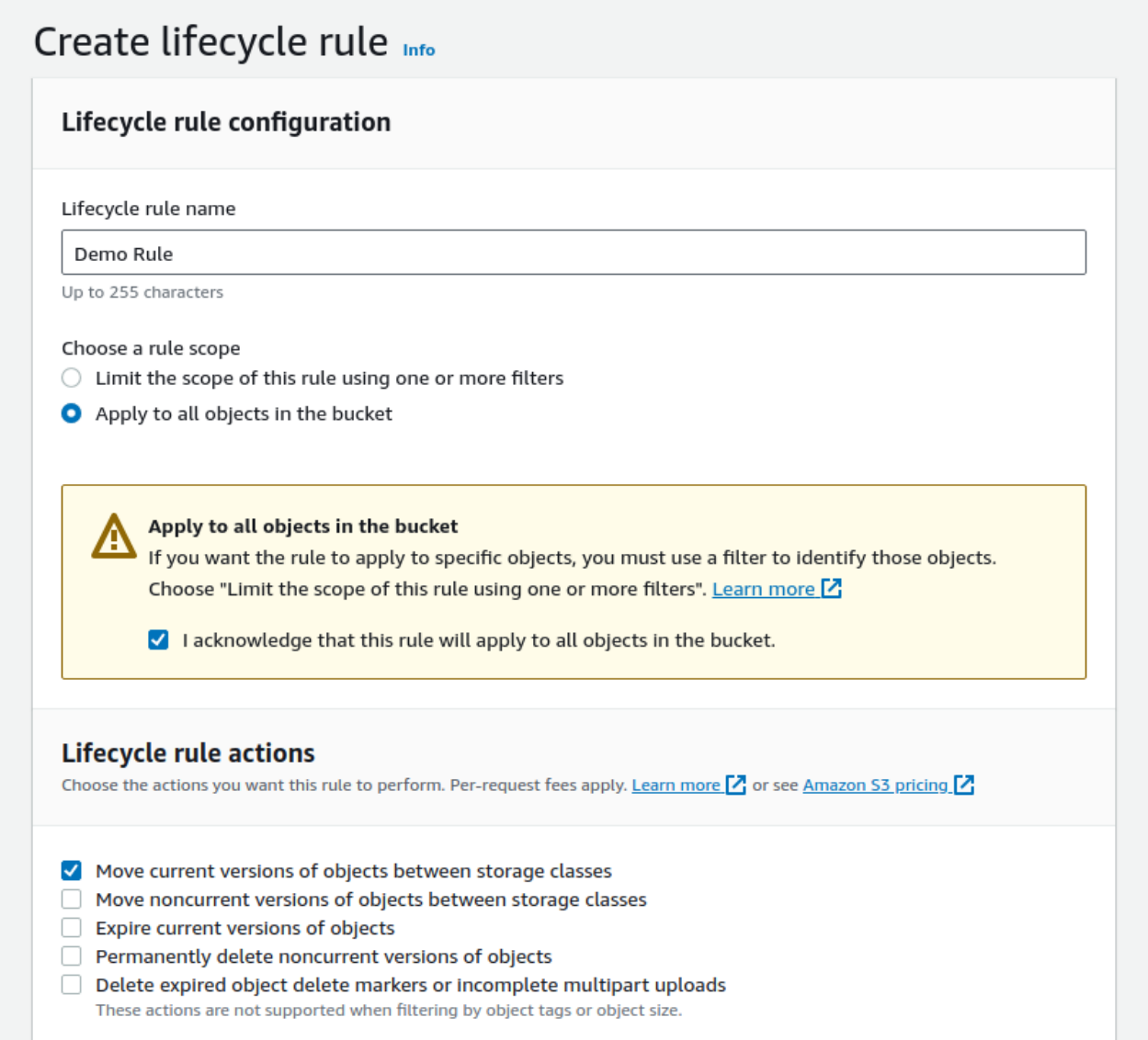
- Choose the Standard-IA option and upload the object.

☐	Name ▲	Type ▽	Last modified ▽	Size ▽	Storage class ▽
☐	coffee.jpg	jpg	November 18, 2023, 14:31:13 (UTC+05:30)	108.4 KB	Standard-IA

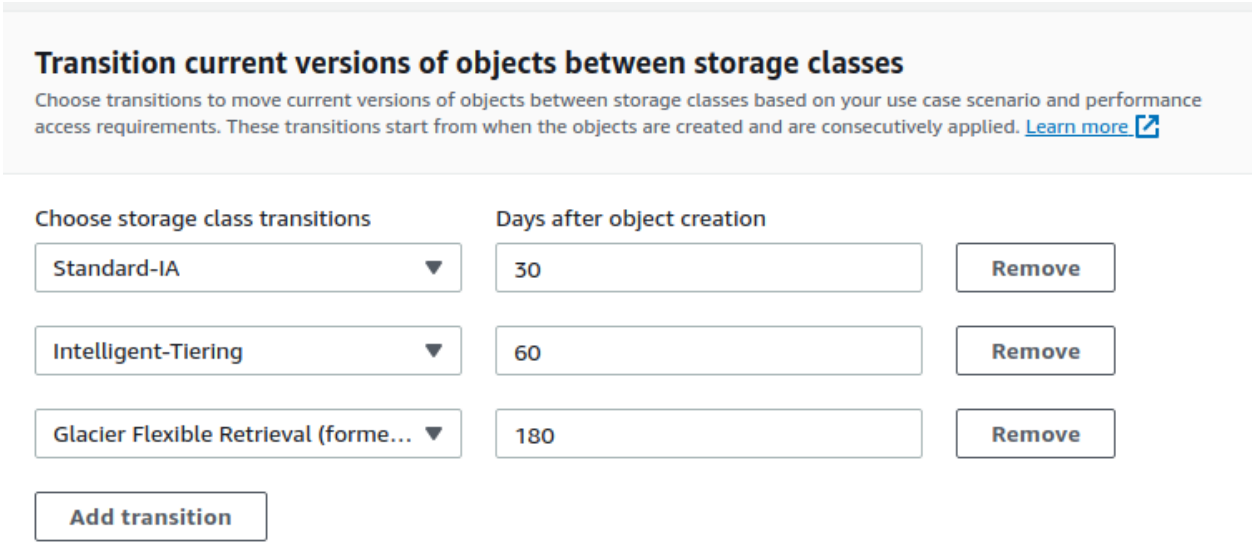
- You can always change the storage class by going to the object >> properties >> storage class >> edit



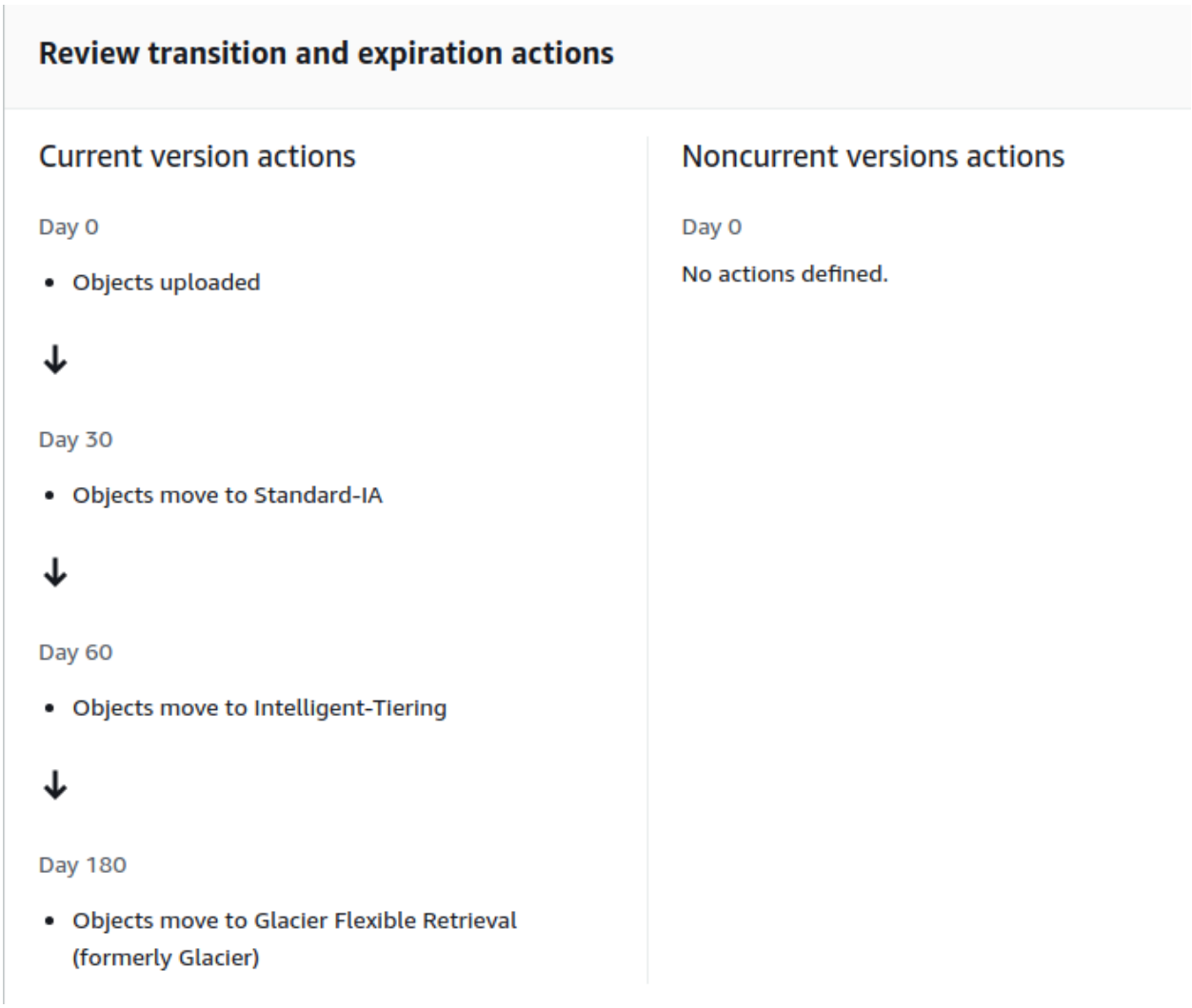
- Let’s now see how we can automate the movement of our objects between different storage classes
- Set the storage class of the object to Intelligent Tiering
- Bucket >> management >> create lifecycle rule



- You can set rules and associate the movement of objects to particular storage classes after x days. These are called transitions.



- Next, you can review your transitions in a roadmap sort of manner.



S3 Encryption

1. Server-side encryption (default)
    - This is always enabled by aws for an added layer of protection.
    - Whenever the object is uploaded by a user, the server encrypts it before putting it into the bucket.
2. Client-side encryption
    - The user encrypts the file before uploading it to the bucket.

IAM Access Analyzer for S3

- A monitoring feature for S3 buckets which ensures only the intended people have access to your S3 bucket.
- It does so by analysing bucket policies, S3 ACLs, access points, etc. and then draws up conclusions as to which buckets are publically accessible, which have been shared with other aws accounts etc. and then you can review it and check if it was expected or maybe if there are any loopholes or accidental rules you might have set and so you can take action.
- This whole thing is powered by IAM Access Analyzer which basically allows you to find out resources in your account that are being shared with other entities.

Shared Responsibility Model for S3

- Infrastructure (global security, durability, availability, sustain 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

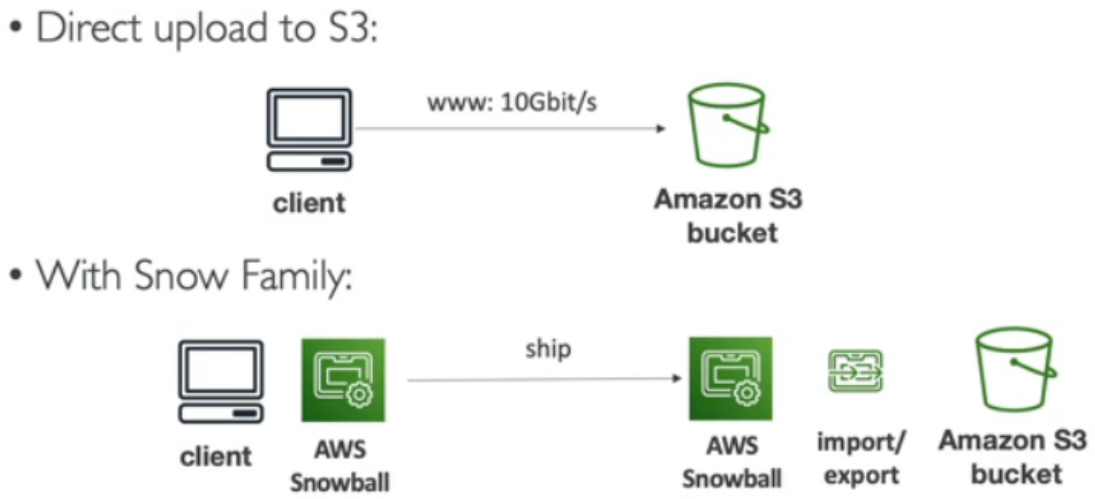
AWS SNOW FAMILY (imp)

- The snow family is a highly secure and portable set of devices which serve two purposes - **data migration and edge computing**

- **DATA MIGRATION**
  - Migration of a lot of data through the network takes a lot of time (in days or months).
  - Moreover, transferring huge chunks data through the network involves more challenges like limited connectivity, limited bandwidth, a shared bandwidth, stability of connection, etc.
  - So, we turn to the Snow Family for easier and quicker migration of data.

  ** If it takes over a week to transfer data over the network, use snowball devices.

For businesses, what this could look like is: you want to put 70 TB of data on aws, then, aws will send you physical offline snowball devices through post and you can upload your data onto those devices and send them back to aws.



- **Devices used for data migration:**
  1. 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

  2. Snowcone & Snowcone SSD (HDD and SSD are available, opt for SSD if you need a faster disk)

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

- Can be sent back to AWS offline, or connect it to internet and use AWS DataSync to send data

3. Snowmobile (IT'S AN ACTUAL TRUCK)

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

https://youtu.be/zX24a71gaVI?si=vIOD9q5Y4fVNikiO THIS IS SO COOL IM CRYING

So, overall

AWS Snow Family for Data Migrations

	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		

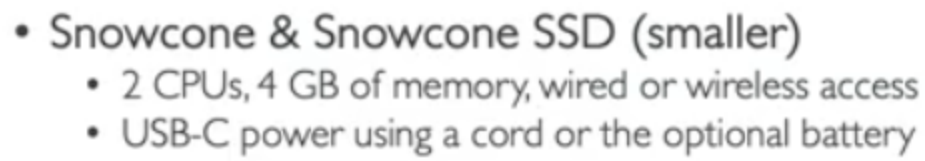
Snow Family – Usage Process

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

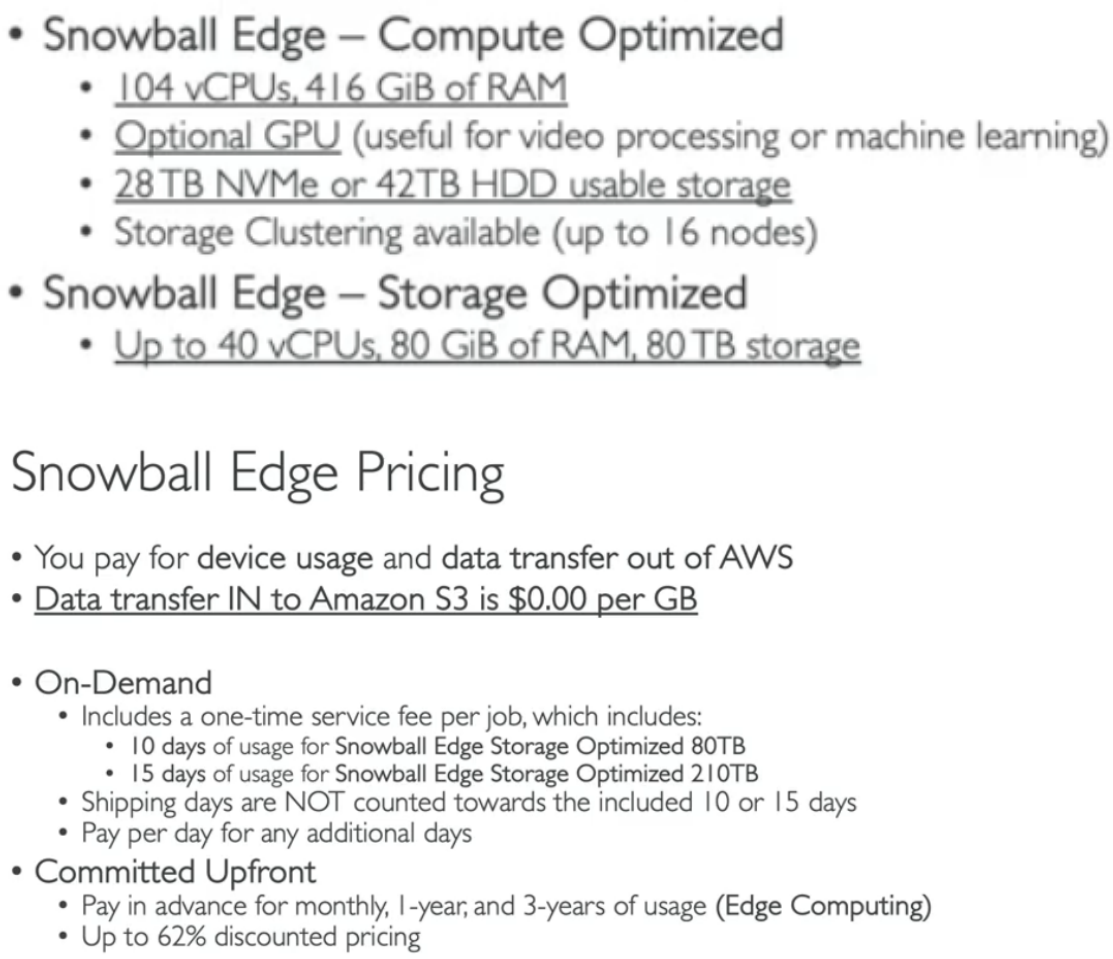
- **EDGE COMPUTING**
  - The purpose is to process data which is being created at edge locations
  - What we basically do is, set up devices to do edge computing (edge computing includes preprocessing data, machine learning, transcoding media streams) and then if needed, we can ship back the device containing processed data to AWS.

- **Devices for edge computing:**

1. Snowcone & Snowcone SSD



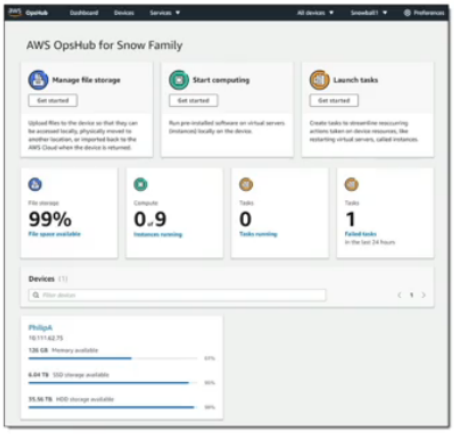
2. Snowball edge



- All these snow devices can run EC2 instance, Lambda functions (using IoT greengrass service)
- All these come with long-term deployment options: 1 or 3 years

- **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-update/

Storage Gateway

- To expose your on-premises data to aws s3, you have to use a storage gateway. This is especially used in case of hybrid cloud models.
- So, the storage gateway acts as a bridge between the on-premises data and cloud data in s3.
- Businesses could opt for storage on the cloud for reasons like disaster recovery, backup or even tiered storage.
- There are 3 types of storage gateways:
  - File Gateway
  - Volume Gateway
  - Tape Gateway


Files
Volumes
Tapes
AWS Storage Gateway
Amazon EBS
S3
Glacier

Good job!
Snowball Edge is best-suited to move petabytes of data and offers computing capabilities. Be careful, it's recommended to use a fleet of Snowballs to move less than 10PBs of data. Over this quantity, it's better-suited to use Snowmobile.

This was discussed in Lecture 86: **AWS Snow Family Overview** >

Question 3:

Which of the following services is a petabyte-scale data moving service (as a fleet) in or out of AWS with computing capabilities?

- ○ Snowcone
- ◉ **Snowball Edge**
- ○ Snowmobile

Good job!
Access Keys are used to sign programmatic requests to the AWS CLI or AWS API.

Question 5:

What are Objects NOT composed of?

- ○ Key
- ○ Value
- ◉ **Access Keys**
- ○ Metadata

Good job!
AWS Snowcone is a small, portable, rugged, and secure edge computing and data transfer device. It provides up to 8 TB of usable storage.

Question 7:

A research team deployed in a location with low-internet connection would like to move 5 TBs of data to the Cloud. Which service can it use?

- ○ Storage Gateway
- ○ Snowball Edge
- ◉ **Snowcone**
- ○ OpsHub

Incorrect answer. Please try again.
Snowball Edge Compute Optimized provides powerful computing resources for higher performance workloads such as machine learning, full motion video analysis, analytics, and local computing stacks. Despite providing local computing capacity, AWS Snowball Edge Compute Optimized devices are not best-suited for data transfer, but instead the high performance workloads mentioned above.

This was discussed in Lecture 86: **AWS Snow Family Overview** ›

Question 9:

A non-profit organization needs to regularly transfer petabytes of data to the cloud and to have access to local computing capacity. Which service can help with this task?

- ○ **Snowball Edge - Storage Optimized**
- ○ Snowball Edge - Compute Optimized
- ○ **Snowcone**
- ○ **Snowmobile**

Good job!
Snowball Edge Storage Optimized devices are well suited for large-scale data migrations and recurring transfer workflows, as well as local computing with higher capacity needs.

This was discussed in Lecture 86: **AWS Snow Family Overview** >

Question 9:

A non-profit organization needs to regularly transfer petabytes of data to the cloud and to have access to local computing capacity. Which service can help with this task?

- (●) **Snowball Edge - Storage Optimized**
- ( ) Snowball Edge - Compute Optimized
- ( ) Snowcone
- ( ) Snowmobile

Good job!
Amazon S3 Standard-Infrequent Access allow you to store infrequently accessed data, with rapid access when needed, has a high durability, and is stored in several Availability Zones to avoid data loss in case of a disaster. It can be used to store data for disaster recovery, backups, etc.

Question 10:

Which S3 Storage Class is suitable for less frequently accessed data, but with rapid access when needed, while keeping a high durability and allowing an Availability Zone failure?

- ○ Amazon S3 Standard - General Purpose
- ○ Amazon Glacier
- ○ Amazon S3 One Zone-Infrequent Access
- ◉ **Amazon S3 Standard-Infrequent Access**