



# **Identity Access Management and S3 storage Bucket**

**Class 25**  
**23/8/2025**

# Acknowledgement

**The series of the IT & Japanese language course is  
Supported by AOTS and OEC.**



Ministry of Economy, Trade and Industry



Overseas Employment Corporation

# What you have Learnt Last Week

**We were focused on following points.**

- Usage of control and loop flow statement
- Performing Linear Algebra in Numpy
- Software development Life cycle
- Importance of Security compliance
- Introduction of Bash Scripting, Ansible, docker and docker compose
- API testing with Postman and Introduction of Jira

# What you will Learn Today

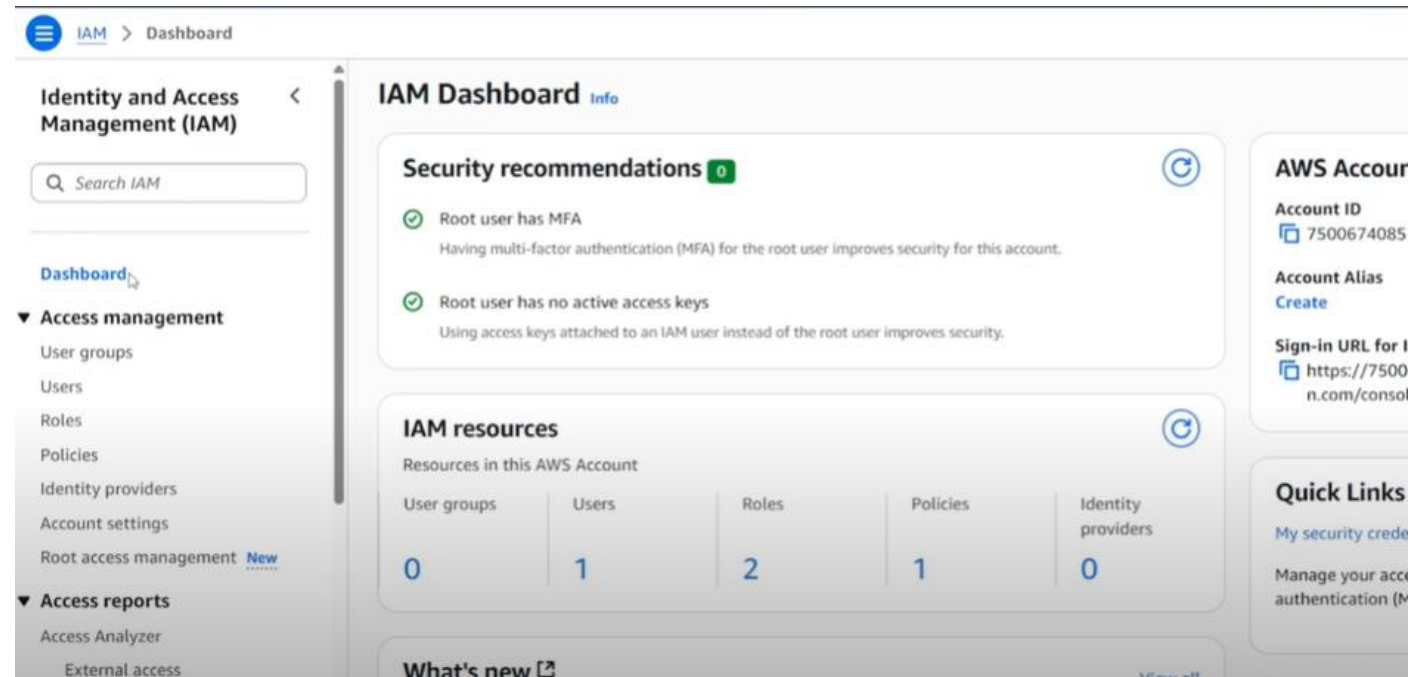
**We will focus on following points.**

1. Introduction to Identity Access Management (IAM) in AWS
2. Setting Up and Configuring S3 Storage Buckets
5. Q&A Session

# Introduction to IAM in AWS

## Foundation of AWS Security

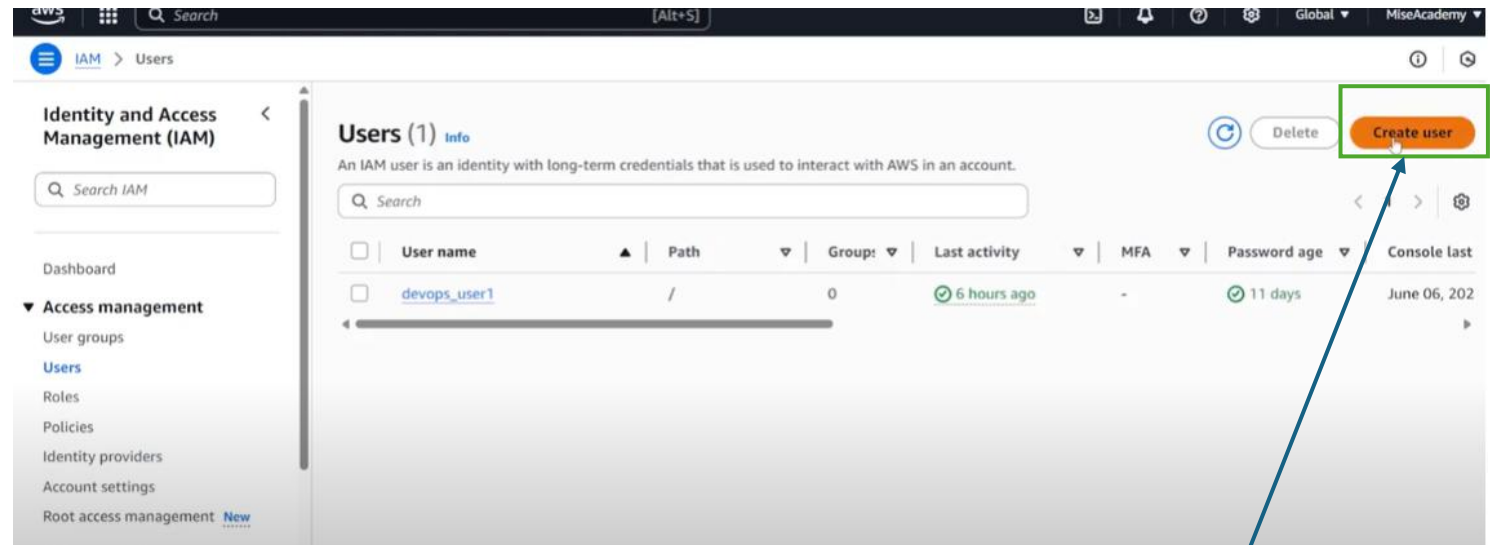
- IAM = Identity and Access Management
- Controls who can access AWS resources and what actions they can perform.
- Central to AWS security model for authentication & authorization.



# IAM Users

## Individual Access Accounts

- Created for real people or applications that need AWS access.
- Each user has unique credentials (password, access keys).
- Best practice: Assign minimum required permissions.



# IAM Groups

## Manage Permissions for Multiple Users

- A group is a collection of IAM users.
- Permissions assigned to a group apply to all its members.
- Example: Developers, Admins, Support.

# IAM Roles

## Temporary Access Without Credentials

- Used by AWS services or federated users.
- No username/password — uses temporary security tokens.
- Examples: EC2 instance accessing S3, cross-account roles.



# IAM Policies

## Permission Blueprints in JSON

- JSON documents that define allow/deny rules.
- Key elements: Version, Statement, Effect, Action, Resource.
- Example:** Allow s3:GetObject for a specific bucket.

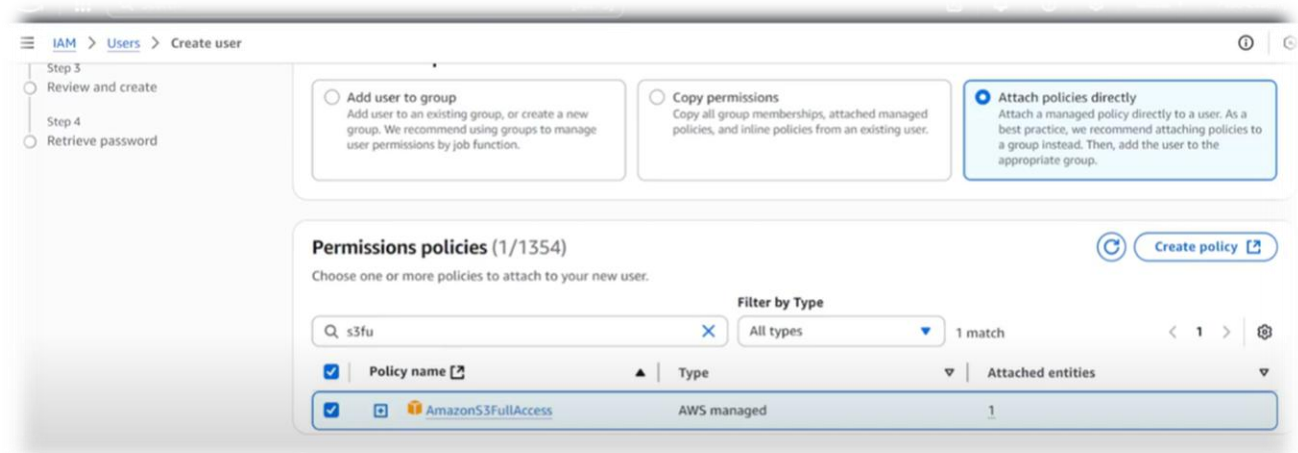
# AWS Managed vs. Customer Managed Policies

## Choosing the Right Policy Type

- **AWS Managed:** Pre-built by AWS, maintained automatically.

- **Customer Managed:** Created & fully controlled by you.

- Best practice: Start with AWS managed → refine into custom.



# Principle of Least Privilege

## **Give Only What's Needed**

- Start with no permissions → add only necessary ones.
- Regularly review & remove unused permissions.
- Prevents accidental or malicious misuse.

# Multi-Factor Authentication (MFA)

## Extra Layer of Security

- Requires password + one-time code from device/app.
- Types: Virtual MFA (Google Authenticator), Hardware MFA, SMS MFA.
- Best practice: Enable MFA for all IAM users with console access.

# IAM Password Policy

## Enforcing Strong Passwords

- Configure length, complexity, rotation.
- Enforce password expiration & prevent reuse.
- Example:** Set password policy to 12 chars, require uppercase/lowercase/numbers.

# IAM Access Keys

## Secure Programmatic Access

- For AWS CLI/SDK/API access.
- Rotate regularly and never hardcode in code.
- Use AWS Secrets Manager or environment variables.
- **Example:** Create an access key, show how to configure in AWS CLI.

# IAM Roles for EC2

## Secure Resource-to-Resource Access

- Assign a role to EC2 → allows it to access AWS services without keys.
- Example: EC2 → S3 backup service.
- Example:** Attach S3 Read-Only role to an EC2 instance.

# Cross-Account Access with IAM Roles

## Cross-Account Access with IAM Roles

- Create a **trust policy** to allow another AWS account to assume the role.
- Useful for multi-account organizations.
- **Example:** Show trust relationship JSON.



# Monitoring IAM with AWS CloudTrail

## Track All IAM Activity

- CloudTrail logs every API call (who, when, what).
- Helps detect suspicious activity & audit compliance.
- **Example:** Open CloudTrail logs and search for “CreateUser” events.

# IAM Security Best Practices

## **Keep Your AWS Secure**

- Enable MFA for all users.
- Rotate keys & passwords regularly.
- Apply least privilege.
- Use IAM roles over access keys whenever possible.
- Monitor with CloudTrail & AWS Config.

# Introduction to Amazon S3

## Scalable Object Storage in AWS

### S3 (Simple Storage Service):

Stores data as objects in buckets.

Global service, accessible via AWS Console, CLI, SDK.

Common use cases: backups, websites, big data storage, media hosting.

### Storage Classes:

- **S3 Standard** – High availability, low latency.
- **S3 Intelligent-Tiering** – Moves objects between storage tiers automatically.
- **S3 Standard-IA** – Lower cost for infrequently accessed data.
- **S3 Glacier / Deep Archive** – Long-term archival storage.

# Creating an S3 Bucket

## Step-by-Step Setup

### Naming Rules:

Globally unique name.

Lowercase letters, numbers, hyphens only.

No spaces or uppercase letters.

### Region Selection:

Choose region closest to your users.

Reduces latency & cost.

The screenshot shows the AWS Management Console interface for creating a new S3 bucket. The page title is 'Create bucket'. The 'AWS Region' is set to 'Europe (Stockholm) eu-north-1'. Under 'Bucket type', the 'General purpose' option is selected. The 'Bucket name' field contains 'myawsbucket'. Below this, there is a section for 'Copy settings from existing bucket - optional' with a 'Choose bucket' button. At the bottom, the 'Object Ownership' section shows 'ACLs disabled (recommended)' selected. A blue arrow points from the 'Region Selection' text above to the 'Europe (Stockholm) eu-north-1' region selection. Another blue arrow points from the 'Naming Rules' text above to the 'Bucket name' input field, which contains 'myawsbucket'.

# Bucket Permissions & ACLs

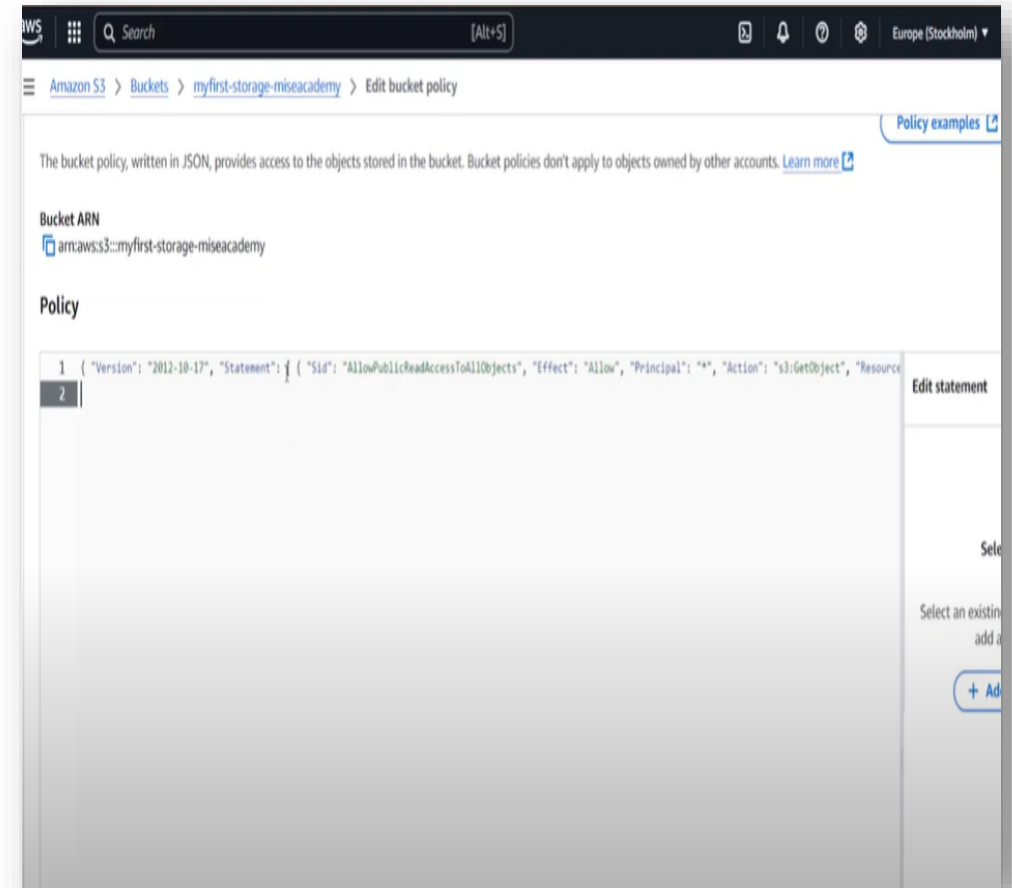
## Controlling Who Can Access Your Data

### ACLs (Access Control Lists):

- Set object/bucket-level permissions.
- Legacy method use policies instead where possible.

### Permissions can be granted to:

- Specific AWS accounts.
- Public (not recommended unless needed).



# Bucket Policies vs. IAM Policies

## Understanding the Difference

**Bucket Policy** → Attached to a bucket (resource-based).

**IAM Policy** → Attached to users, groups, or roles (identity-based).

### Example:

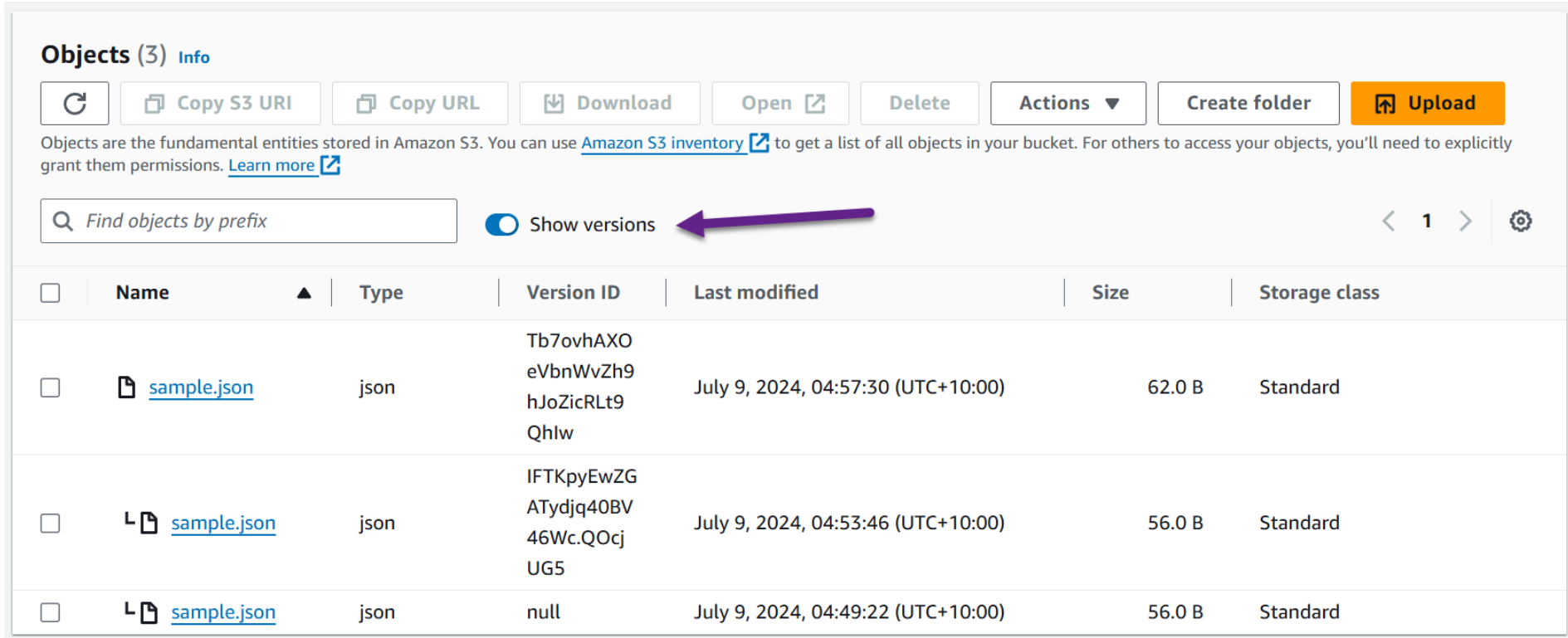
**Bucket Policy:** Allow public read to a bucket.

**IAM Policy:** Allow a user to read from S3 buckets.

# Enabling Versioning

## Backup & Recovery of Objects

- Keeps multiple versions of an object.
- Protects against accidental deletion/overwriting.
- Can increase costs use lifecycle rules to delete old versions.








**Objects (3)** [Info](#)

[Refresh](#) [Copy S3 URI](#) [Copy URL](#) [Download](#) [Open](#) [Delete](#) [Actions](#) [Create folder](#) [Upload](#)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

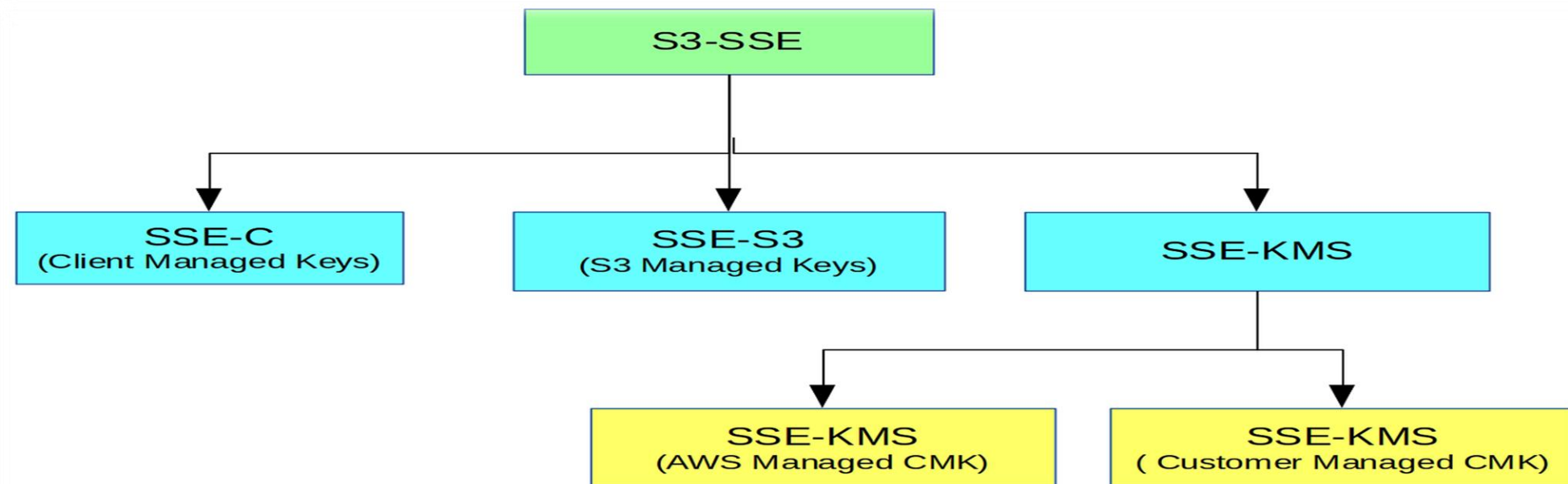
☒ Show versions < 1 > ⚙️

<input type="checkbox"/>	Name ▲	Type	Version ID	Last modified	Size	Storage class
<input type="checkbox"/>	 <a href="#">sample.json</a>	json	Tb7ovhAXO eVbnWvZh9 hJoZicRLt9 Qhlw	July 9, 2024, 04:57:30 (UTC+10:00)	62.0 B	Standard
<input type="checkbox"/>	  <a href="#">sample.json</a>	json	IFTKpyEwZG ATydlq40BV 46Wc.QOcJ UG5	July 9, 2024, 04:53:46 (UTC+10:00)	56.0 B	Standard
<input type="checkbox"/>	  <a href="#">sample.json</a>	json	null	July 9, 2024, 04:49:22 (UTC+10:00)	56.0 B	Standard

# Server-Side Encryption (SSE)

## Keeping Data Secure at Rest

- **SSE-S3:** Managed keys by AWS.
- **SSE-KMS:** Customer managed keys in KMS.
- **SSE-C:** Customer-provided keys (rarely used).





# Public Access Settings

## Controlling Internet Access to Buckets

- Block public access (recommended).
- If public, risk of data leaks.
- Common use case: hosting public static website files.

havecamerawithtravel.developer

Overview Properties Permissions Management Access points

Block public access Access Control List Bucket Policy CORS configuration

**Block public access (bucket settings)**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to all your S3 buckets and objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to your buckets or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

**Block all public access**  
On

Block public access to buckets and objects granted through *new* access control lists (ACLs)  
On

Block public access to buckets and objects granted through *any* access control lists (ACLs)  
On

Block public access to buckets and objects granted through *new* public bucket or access point policies  
On

Block public and cross-account access to buckets and objects through *any* public bucket or access point policies  
On

Edit

# Lifecycle Rules

## Automating Storage Cost Optimization

Move objects between storage classes automatically.

### Common rule:

After 30 days → move to Standard-IA.

After 180 days → move to Glacier.

Also delete old versions/expired files automatically.

Amazon S3 > Buckets > lifecycle-rule-test-54321 > Lifecycle configuration

### Lifecycle configuration [Info](#)

To manage your objects so that they are stored cost effectively throughout their lifecycle, configure their lifecycle. A lifecycle configuration is a set of rules that define actions that Amazon S3 applies to a group of objects. Lifecycle rules run once per day.

#### Lifecycle rules (1)

Use lifecycle rules to define actions you want Amazon S3 to take during an object's lifetime such as transitioning objects to another storage class, archiving them, or deleting them after a specified period of time. [Learn more](#)

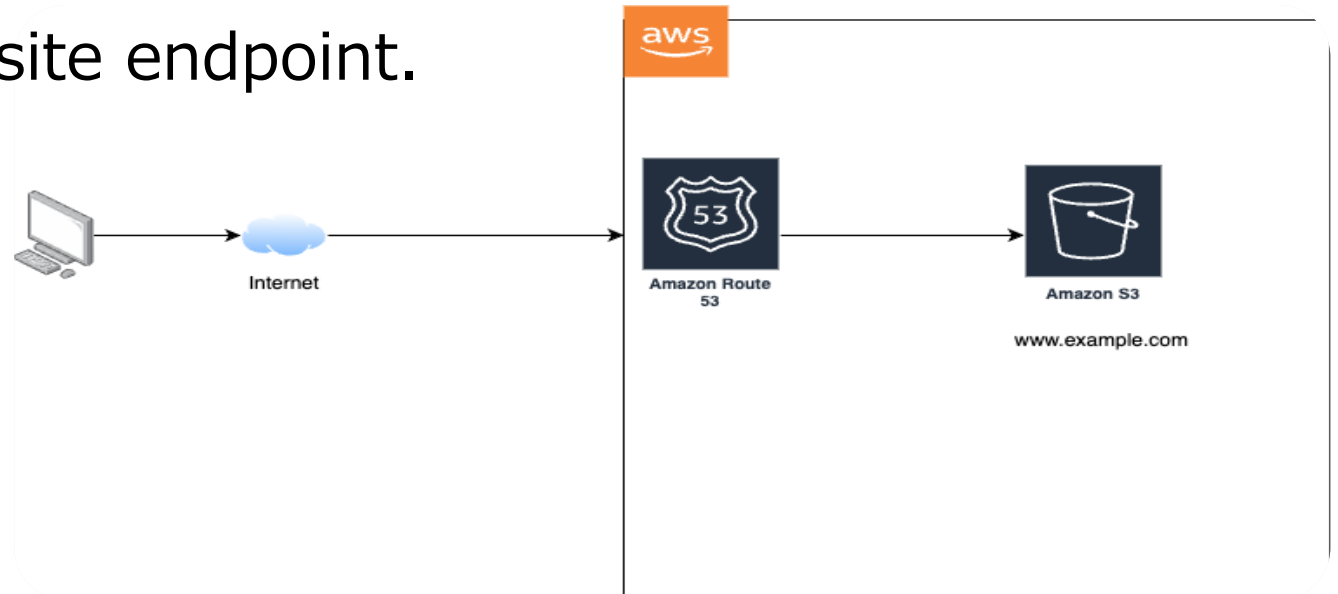
[Refresh](#) [View details](#) [Edit](#) [Delete](#) [Actions](#) [Create lifecycle rule](#)

	Lifecycle rule name	Status	Scope	Current version actions	Noncurrent versions actions	Expired object delete markers	Incomplete uploads
<input type="radio"/>	<a href="#">Glacier Lifecycle Rule Test</a>	Enabled	Filtered	Transition to Glacier Flexible Retrieval (formerly Glacier), then Glacier Deep Archive, then expires	-	-	-

# Hosting a Static Website on S3

## Turn S3 Into a Simple Web Host

- Enable “Static Website Hosting” in bucket properties.
- Upload index.html & error.html.
- Public read access required.
- Access site via provided S3 website endpoint.



# Logging & Monitoring S3 Access

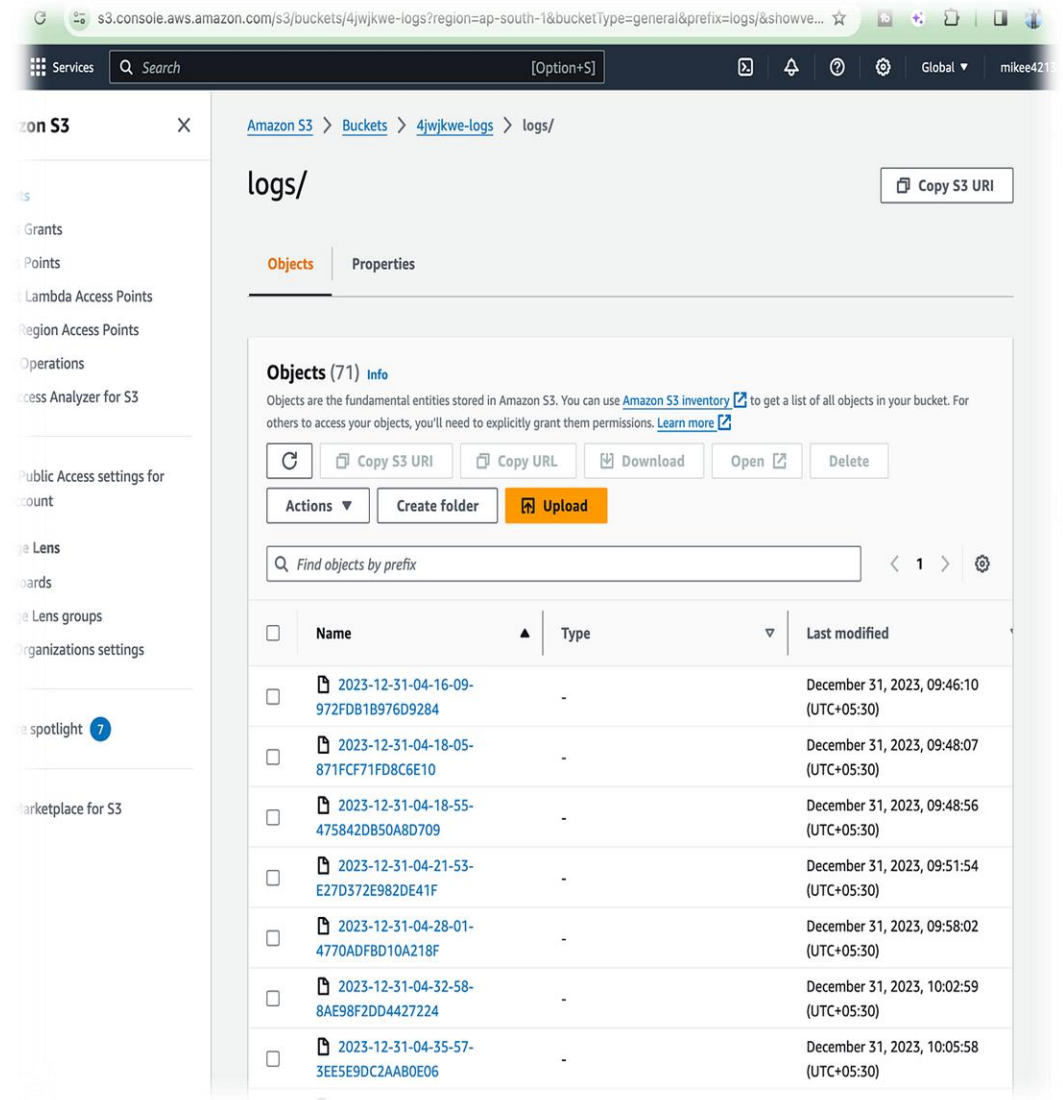
## Tracking Who Accessed What

### S3 Server Access Logs:

- Logs requests made to the bucket.
- Stored in another bucket.

### AWS CloudTrail:

- Logs API calls to S3 (who, when, action).



# Assignment

# Assignment

1. Create an IAM user and then login into AWS with IAM user
2. Create a simple static website and deploy on S3 bucket



# Quiz Section

# Quiz

**Everyone student should click on submit button before time ends otherwise MCQs will not be submitted**

## **[Guidelines of MCQs]**

1. There are 20 MCQs
2. Time duration will be 10 minutes
3. This link will be share on 12:25pm (Pakistan time)
4. MCQs will start from 12:30pm (Pakistan time)
5. This is exact time and this will not change
6. Everyone student should click on submit button otherwise MCQs will not be submitted after time will finish
7. Every student should submit Github profile and LinkedIn post link for every class. It include in your performance



# Assignment

**Assignment should be submit before the next class**

## **[Assignments Requirements]**

1. Create a post of today's lecture and post on LinkedIn.
2. Make sure to tag @Plus W @Pak-Japan Centre and instructors LinkedIn profile
3. Upload your code of assignment and lecture on GitHub and share your GitHub profile in respective your region group WhatsApp group
4. If you have any query regarding assignment, please share on your region WhatsApp group.
5. Students who already done assignment, please support other students

# Q&A Session

ありがとうございます。

Thank you.

شكريا



For the World with Diverse Individualities