AWS S3

Amazon S3 Security & Access Control

S3 Security Features

- **IAM Policies**: Grants permissions based on AWS IAM roles and policies.
- **Bucket Policies**: Defines access rules at the bucket level.
- **ACLs (Access Control Lists)**: Grants specific permissions to AWS accounts.
- **S3 Block Public Access**: Prevents accidental data exposure to the public.

IAM Role-Based Access

- Use IAM roles to grant temporary permissions.
- Policies define **Allow/Deny** actions.
- Example policy to allow read-only access to a bucket:

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

Encryption in S3

Types of Encryption

1. **Server-Side Encryption (SSE)**
   - **SSE-S3**: AWS manages the encryption keys.
   - **SSE-KMS**: Uses AWS KMS for key management.
   - **SSE-C**: Customer provides their own encryption keys.
2. **Client-Side Encryption**
   - Data is encrypted before being uploaded to S3.

Enabling Default Encryption for a Bucket

1. Go to **S3 Console** → Select a bucket.
2. Click **Properties** → Enable **Default Encryption**.
3. Choose **SSE-S3** or **SSE-KMS**.

S3 Versioning & Object Lock

S3 Versioning

- Keeps multiple versions of an object.
- Helps recover from unintended deletions or changes.

Object Lock & Retention

- **Prevents deletion/modification of objects** for a defined period.
- Two modes:
  - **Governance Mode**: Users with special permissions can delete objects.
  - **Compliance Mode**: No one, even the root user, can delete objects.

Summary

- Implement IAM roles and bucket policies to secure access.
- Use encryption (SSE-S3, SSE-KMS, SSE-C) to protect data.
- Enable versioning and object lock for data integrity.
- Utilize MFA Delete and bucket policies to prevent unauthorized access.

Next Steps

- Learn about **S3 Performance Optimization & Data Management**
- Explore **Lifecycle Policies, Replication, and S3 Transfer Acceleration**.