AWS S3

Simple Storage Service

Complete Guide to AWS Object Storage

Prepared By: Rashi Rana
AWS Corporate Trainer

Table of Contents

📦 S3 Fundamentals

🗂️ Storage Classes

🔄 Advanced Features

🔐 Security & Best Practices

Prepared By: Rashi Rana
AWS Corporate Trainer

What is Amazon S3?

Amazon Simple Storage Service (S3) is an object storage service that offers industry-leading scalability, data availability, security, and performance.

Key Benefits

- **Scalability:** Store virtually unlimited amounts of data
- **Durability:** 99.999999999% (11 9's) durability
- **Availability:** 99.99% availability SLA
- **Security:** Comprehensive security and compliance capabilities
- **Performance:** High transfer speeds and low latency

Core Concepts

🪣 Buckets

Containers for objects with globally unique names

📄 Objects

Files stored in buckets (up to 5TB each)

🔑 Keys

Unique identifier for objects within a bucket

🌍 Regions

Geographic location where buckets are stored

```
# S3 URL Structure
https://bucket-name.s3.region.amazonaws.com/object-key
https://my-website-bucket.s3.us-east-1.amazonaws.com/images/logo.png

# Virtual Hosted Style (Recommended)
https://bucket-name.s3.amazonaws.com/object-key

# Path Style (Legacy)
https://s3.amazonaws.com/bucket-name/object-key
```

Prepared By: Rashi Rana
AWS Corporate Trainer

S3 Naming Conventions

Bucket Naming Rules

✅ Requirements

- 3-63 characters long
- Lowercase letters, numbers, hyphens only
- Must start and end with letter or number
- Globally unique across all AWS accounts
- No consecutive periods or hyphens
- Cannot be formatted as IP address

Valid Examples	Invalid Examples	Reason
my-company-logs	My-Company-Logs	Contains uppercase letters
website-assets-2024	website_assets_2024	Contains underscores
backup-data-us-east	backup..data	Consecutive periods
app-config-prod	192.168.1.1	IP address format

Object Key Best Practices

📁 Logical Structure

- Use forward slashes for hierarchy
- year/month/day/file.txt
- department/project/version/

🔤 Character Guidelines

- Avoid special characters
- Use hyphens instead of spaces
- Consider URL encoding

Prepared By: Rashi Rana
AWS Corporate Trainer

S3 Storage Classes

S3 offers multiple storage classes designed for different use cases, providing cost optimization based on access patterns and performance requirements.

Storage Class	Use Case	Availability	Min Storage Duration	Retrieval Fee
S3 Standard	Frequently accessed data	99.99%	None	None
S3 Standard-IA	Infrequently accessed data	99.9%	30 days	Per GB retrieved
S3 One Zone-IA	Infrequent access, single AZ	99.5%	30 days	Per GB retrieved
S3 Glacier Instant	Archive with instant retrieval	99.9%	90 days	Per GB retrieved
S3 Glacier Flexible	Archive with flexible retrieval	99.99%	90 days	Per request + GB
S3 Glacier Deep Archive	Long-term archive	99.99%	180 days	Per request + GB

🚀 Standard Classes

- **Standard:** General purpose
- **Standard-IA:** Backup, disaster recovery
- **One Zone-IA:** Secondary backups

🧊 Glacier Classes

- **Instant:** Millisecond retrieval
- **Flexible:** 1-12 hours retrieval
- **Deep Archive:** 12-48 hours retrieval

Prepared By: Rashi Rana
AWS Corporate Trainer

S3 Versioning

S3 Versioning allows you to keep multiple variants of an object in the same bucket, providing protection against accidental deletion or modification.

Key Features

- **Multiple Versions:** Store multiple versions of the same object
- **Version ID:** Unique identifier for each object version
- **Current Version:** Latest version retrieved by default
- **Delete Protection:** Prevents permanent data loss

Versioning States

🔴 Unversioned (Default)

- No versioning enabled
- Objects overwrite each other
- No version ID assigned

🟢 Versioning Enabled

- New versions created on upload
- Unique version IDs assigned
- Previous versions preserved

🟡 Versioning Suspended

- No new versions created
- Existing versions preserved
- New objects get null version ID

```
# Enable versioning on bucket
aws s3api put-bucket-versioning \
    --bucket my-bucket \
    --versioning-configuration Status=Enabled

# List object versions
aws s3api list-object-versions \
    --bucket my-bucket \
    --prefix path/to/object

# Get specific version
aws s3api get-object \
    --bucket my-bucket \
    --key myfile.txt \
    --version-id "version-id-here" \
    myfile-v1.txt
```

⚠️ Important Considerations

- Versioning cannot be disabled, only suspended
- Each version is billed as a separate object
- Delete operations create delete markers
- Use lifecycle policies to manage old versions

Prepared By: Rashi Rana
AWS Corporate Trainer

S3 Lifecycle Rules

Lifecycle rules automatically transition objects between storage classes or delete them based on predefined criteria, helping optimize costs and manage data retention.

Lifecycle Actions

- **Transition Actions:** Move objects to different storage classes
- **Expiration Actions:** Delete objects after specified time
- **Incomplete Multipart Upload:** Clean up failed uploads
- **Previous Versions:** Manage non-current object versions

Common Lifecycle Patterns

📊 Data Archival

- Day 0: S3 Standard
- Day 30: S3 Standard-IA
- Day 90: S3 Glacier Flexible
- Day 365: S3 Glacier Deep Archive

🗑️ Log Management

- Day 0: S3 Standard
- Day 7: S3 Standard-IA
- Day 30: S3 Glacier Flexible
- Day 90: Delete

```
# Example Lifecycle Configuration
```

```
{
    "Rules": [
        {
            "ID": "DataArchivalRule",
            "Status": "Enabled",
            "Filter": {
                "Prefix": "documents/"
            },
            "Transitions": [
                {
                    "Days": 30,
                    "StorageClass": "STANDARD_IA"
                },
                {
                    "Days": 90,
                    "StorageClass": "GLACIER"
                },
                {
                    "Days": 365,
                    "StorageClass": "DEEP_ARCHIVE"
                }
            ],
            "Expiration": {
                "Days": 2555
            }
        }
    ]
}
```

✅ Best Practices

- Use prefixes to target specific object groups
- Consider minimum storage duration charges
- Test lifecycle rules on non-production data first
- Monitor lifecycle rule performance with CloudWatch

Prepared By: Rashi Rana
AWS Corporate Trainer

S3 Access Control

S3 provides multiple layers of access control to secure your data, from bucket-level policies to object-level permissions.

Access Control Methods

Method	Scope	Use Case	Granularity
IAM Policies	User/Role based	Control who can access S3	User/Group level
Bucket Policies	Bucket level	Cross-account access, public access	Bucket/Object level
ACLs	Bucket/Object level	Simple permissions (legacy)	Basic read/write
Access Points	Application level	Simplified access management	Application specific

Public vs Private Access

🔒 Private Access (Default)

- Only bucket owner has access
- Requires authentication

🌐 Public Access

- Accessible via internet
- No authentication required
- Use for static websites

- IAM policies control access
- Secure by default

- Requires explicit configuration

```
# Block all public access (recommended)
aws s3api put-public-access-block \
    --bucket my-bucket \
    --public-access-block-configuration \

BlockPublicAcls=true,IgnorePublicAcls=true,BlockPublicPolicy
=true,RestrictPublicBuckets=true

# Example bucket policy for public read access
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "PublicReadGetObject",
            "Effect": "Allow",
            "Principal": "*",
            "Action": "s3:GetObject",
            "Resource": "arn:aws:s3:::my-public-bucket/*"
        }
    ]
}
```

⚠ Security Best Practices

- Enable Block Public Access by default
- Use least privilege principle
- Regularly audit bucket policies
- Enable CloudTrail for access logging

S3 Encryption

S3 provides multiple encryption options to protect your data at rest and in transit, ensuring comprehensive data security.

Server-Side Encryption (SSE)

Encryption Type	Key Management	Use Case	Header
SSE-S3	AWS managed keys	Default encryption	x-amz-server-side-encryption: AES256
SSE-KMS	AWS KMS keys	Audit trail, key rotation	x-amz-server-side-encryption: aws:kms
SSE-C	Customer provided keys	Full key control	x-amz-server-side-encryption-customer-algorithm
DSSE-KMS	Dual layer KMS	Enhanced security	x-amz-server-side-encryption: aws:kms:dsse

Client-Side Encryption

🔐 CSE-KMS

- AWS KMS managed keys
- Encrypt before upload

🔑 CSE-C

- Customer managed keys
- Full control over encryption

- AWS SDK handles encryption

- Application handles encryption

```
# Enable default bucket encryption (SSE-S3)
aws s3api put-bucket-encryption \
    --bucket my-bucket \
    --server-side-encryption-configuration '{
        "Rules": [
            {
                "ApplyServerSideEncryptionByDefault": {
                    "SSEAlgorithm": "AES256"
                }
            }
        ]
    }'

# Upload with SSE-KMS encryption
aws s3 cp myfile.txt s3://my-bucket/ \
    --sse aws:kms \
    --sse-kms-key-id alias/my-key
```

✅ Encryption Best Practices

- Enable default bucket encryption
- Use SSE-KMS for audit requirements
- Enforce encryption in bucket policies
- Use HTTPS for data in transit

S3 Replication

S3 Replication automatically copies objects across buckets in the same or different AWS regions, providing data redundancy and compliance capabilities.

Replication Types

🌍 Cross-Region Replication (CRR)

- Replicate to different AWS regions
- Compliance and data sovereignty
- Disaster recovery
- Reduce latency for global users

🏢 Same-Region Replication (SRR)

- Replicate within same region
- Log aggregation
- Production to test replication
- Data backup and archival

Replication Requirements

Prerequisites

- **Versioning:** Must be enabled on both source and destination buckets
- **IAM Role:** S3 needs permissions to replicate objects
- **Different Buckets:** Source and destination must be different buckets

- **Ownership:** Source bucket owner must have permissions

Feature	What Replicates	What Doesn't Replicate
Objects	New objects after rule creation	Existing objects (use S3 Batch Replication)
Metadata	Object metadata and tags	Bucket-level settings
Encryption	SSE-S3, SSE-KMS, SSE-C	Unencrypted to encrypted (configurable)
Deletions	Delete markers (optional)	Permanent deletions of versions

```
# Create replication configuration
{
    "Role": "arn:aws:iam::account:role/replication-role",
    "Rules": [
        {
            "ID": "ReplicateToBackup",
            "Status": "Enabled",
            "Filter": {
                "Prefix": "documents/"
            },
            "Destination": {
                "Bucket": "arn:aws:s3:::backup-bucket",
                "StorageClass": "STANDARD_IA"
            }
        }
    ]
}

# Apply replication configuration
aws s3api put-bucket-replication \
    --bucket source-bucket \
    --replication-configuration file://replication.json
```

Prepared By: Rashi Rana
AWS Corporate Trainer

S3 Best Practices

🔐 Security

- Enable Block Public Access by default
- Use IAM policies and bucket policies
- Enable default encryption
- Enable access logging
- Use MFA Delete for critical buckets
- Regular access reviews

💰 Cost Optimization

- Use appropriate storage classes
- Implement lifecycle policies
- Delete incomplete multipart uploads
- Monitor storage usage with CloudWatch
- Use S3 Storage Class Analysis
- Consider S3 Intelligent-Tiering

⚡ Performance

- Use random prefixes for high request rates
- Enable Transfer Acceleration
- Use multipart upload for large files
- Implement retry logic with exponential backoff
- Use CloudFront for global distribution

🛡️ Reliability

- Enable versioning for critical data
- Set up cross-region replication
- Use multiple storage classes
- Implement backup strategies
- Monitor with CloudWatch alarms

- Optimize request patterns

- Test disaster recovery procedures

S3 Request Rate Guidelines

- **GET/HEAD/DELETE:** 5,500 requests per second per prefix
- **PUT/COPY/POST:** 3,500 requests per second per prefix
- **LIST:** No specific limit, but consider pagination
- **Multipart Upload:** 10,000 parts per upload

```
# S3 Performance Best Practices Commands

# Enable Transfer Acceleration
aws s3api put-bucket-accelerate-configuration \
    --bucket my-bucket \
    --accelerate-configuration Status=Enabled

# Multipart upload for large files
aws s3 cp largefile.zip s3://my-bucket/ \
    --storage-class STANDARD_IA \
    --metadata key1=value1,key2=value2

# Sync with delete (be careful!)
aws s3 sync ./local-folder s3://my-bucket/folder/ \
    --delete \
    --exclude "*.tmp"
```

Prepared By: Rashi Rana
AWS Corporate Trainer

S3 Monitoring & Troubleshooting

Key Metrics to Monitor

📊 Storage Metrics

- BucketSizeBytes
- NumberOfObjects
- Storage class distribution
- Lifecycle transitions

🔄 Request Metrics

- AllRequests
- GetRequests
- PutRequests
- DeleteRequests

⚠️ Error Metrics

- 4xxErrors
- 5xxErrors
- FirstByteLatency
- TotalRequestLatency

💸 Cost Metrics

- Data transfer costs
- Request costs
- Storage costs by class
- Lifecycle transition costs

⚠️ Common Issues & Solutions

- **403 Forbidden:** Check IAM policies, bucket policies, and ACLs
- **404 Not Found:** Verify bucket name, region, and object key
- **503 Slow Down:** Implement exponential backoff and retry logic
- **High Costs:** Review storage classes and lifecycle policies

- **Slow Performance:** Check request patterns and use CloudFront

✅ Monitoring Tools

- **CloudWatch:** Metrics, alarms, and dashboards
- **CloudTrail:** API call logging and auditing
- **Access Logs:** Detailed request logging
- **Cost Explorer:** Cost analysis and optimization
- **S3 Storage Lens:** Organization-wide storage analytics

Prepared By: Rashi Rana
AWS Corporate Trainer

Summary

Key Takeaways

- **S3 Fundamentals:** Object storage with buckets, objects, and keys
- **Storage Classes:** Choose based on access patterns and cost requirements
- **Versioning:** Protect against accidental deletion and modification
- **Lifecycle Rules:** Automate cost optimization and data management
- **Security:** Multiple layers of access control and encryption
- **Replication:** Ensure data redundancy and compliance

Next Steps

🧪 Hands-on Practice

- Create and configure S3 buckets
- Set up lifecycle policies
- Configure replication rules

📚 Advanced Topics

- S3 Event Notifications
- S3 Select and Glacier Select
- S3 Batch Operations
- S3 Multi-Region Access Points

- Test different storage classes

💡 Remember

S3 is designed for 99.999999999% (11 9's) durability and 99.99% availability. It's not just storage - it's a platform for building scalable, reliable applications.

Thank You!

Questions & Discussion

Prepared By: Rashi Rana
AWS Corporate Trainer