Step-by-Step Implementation

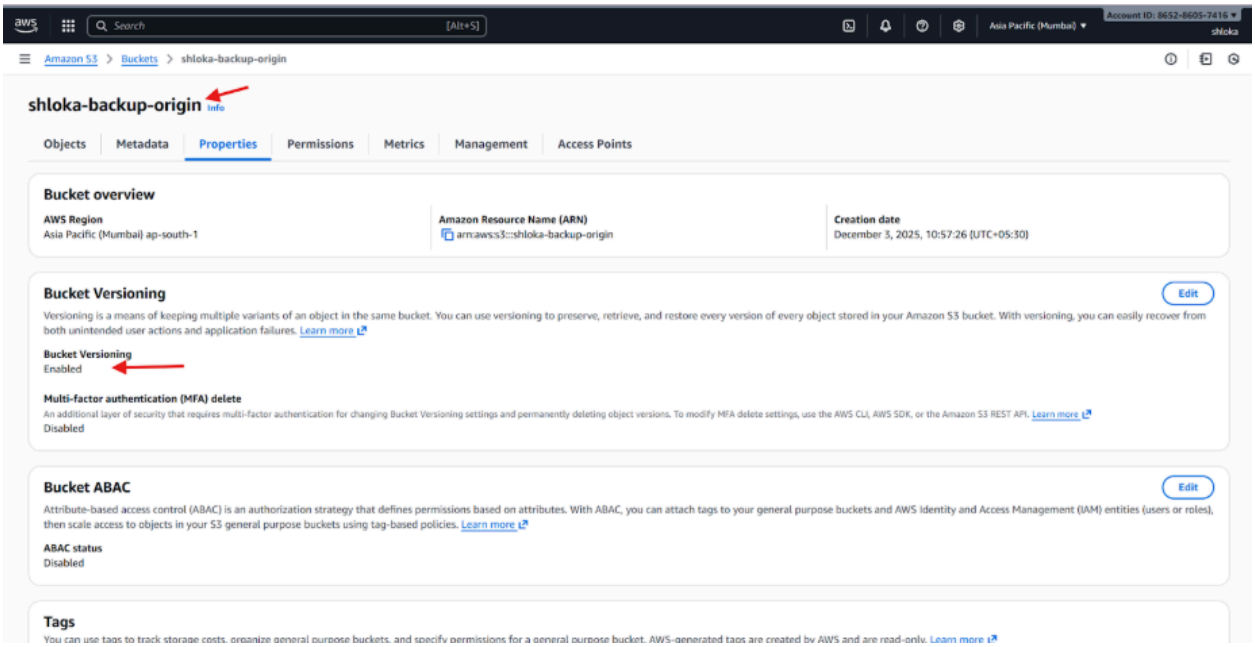
Step 1: Primary Bucket Creation

What was done:

- Created S3 bucket: shloka-backup-origin in ap-south-1
- Enabled Block all public access
- Enabled Versioning during Bucket Creation
- Configured SSE-S3 encryption

Why was it done

Versioning is required to maintain object history for replication functionality. The primary bucket serves as the source of truth for backup data with all security measures in place.



Step 2: Destination Bucket Creation (N. Virginia)

What was done:

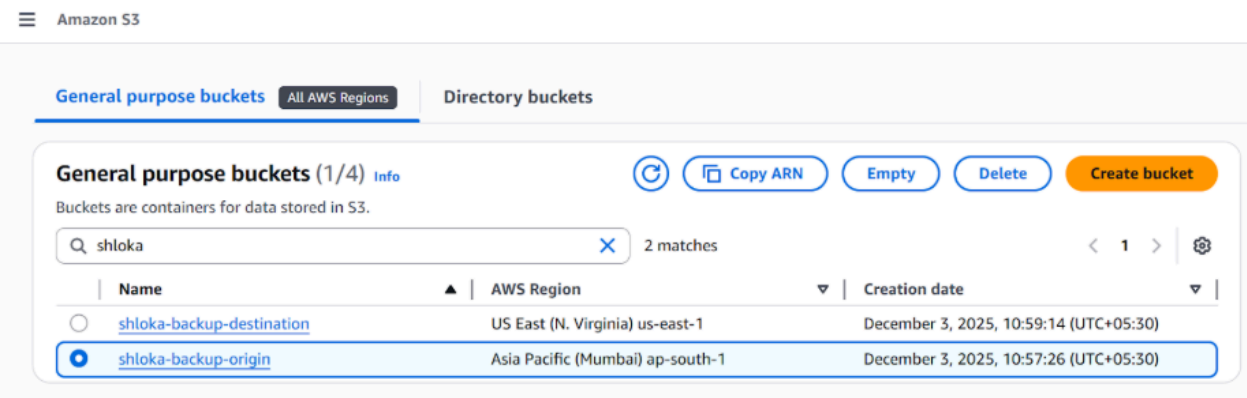
- Created S3 bucket: shloka-backup-destination in us-east-1

- Enabled Block all public access
- Enabled Versioning during Bucket Creation
- Configured SSE-S3 encryption

Why was it done

Geographic diversity between Mumbai (ap-south-1) and N. Virginia (us-east-1) ensures that regional failures do not impact data availability. Cross-region replication achieves 11 9's durability.

Buckets created:



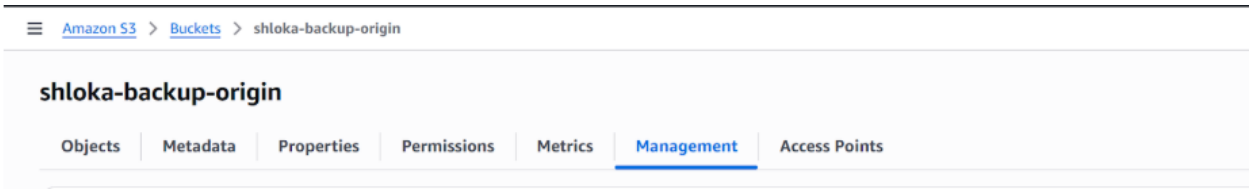
Step 3: Replication Rule Configuration

What was done:

- Created replication rule: prod-objects-replication
- Configured tag-based filter: env=prod
- Enabled delete marker replication
- Configured for existing objects
- Checked auto assign IAM role

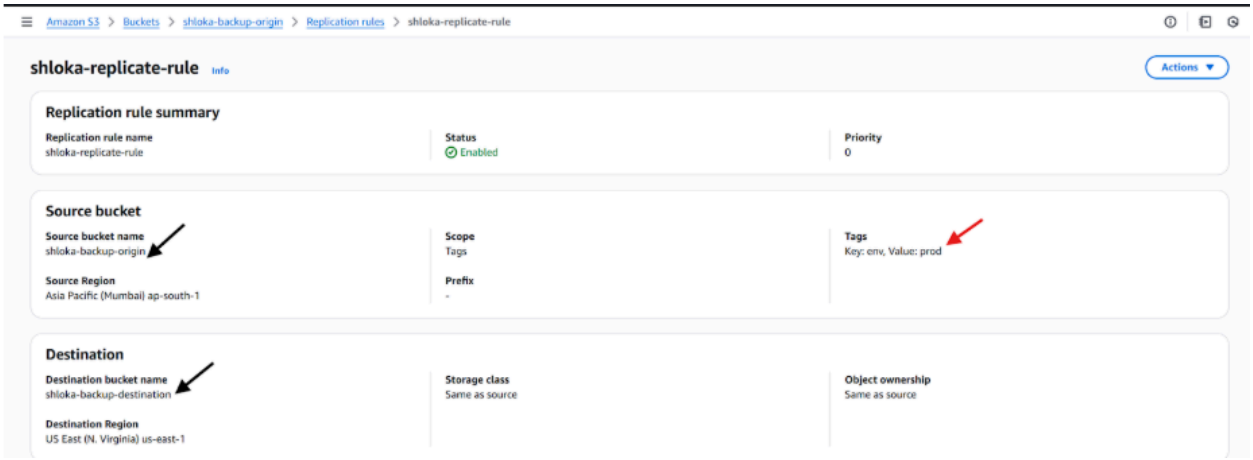
How was it done

- Navigated through the console
- Amazon S3 → Origin Bucket → Management → Replication rule



Why was it done

- Tag-based filtering ensures only production objects replicate, reducing costs while maintaining critical data protection. This selective approach optimizes both storage and bandwidth



Step 4: Bucket Lifecycle Configuration

What was done:

- Created lifecycle rule: intelligent-tiering-transition
- Configured 60-day transition to INTELLIGENT_TIERING
- Configured 30-day noncurrent version expiration

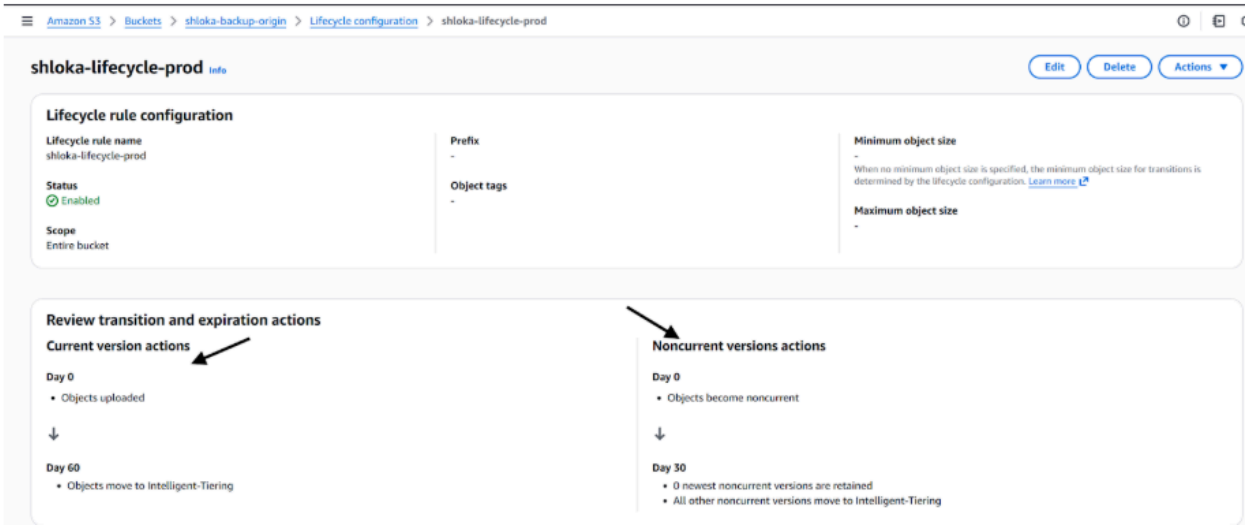
How was it done

- Navigated through the console
- Amazon S3 → Origin Bucket → Management → Life Cycle

Why was it done

- These tiered actions optimize costs while maintaining data availability and compliance. The 60-day transition to INTELLIGENT_TIERING reflects typical

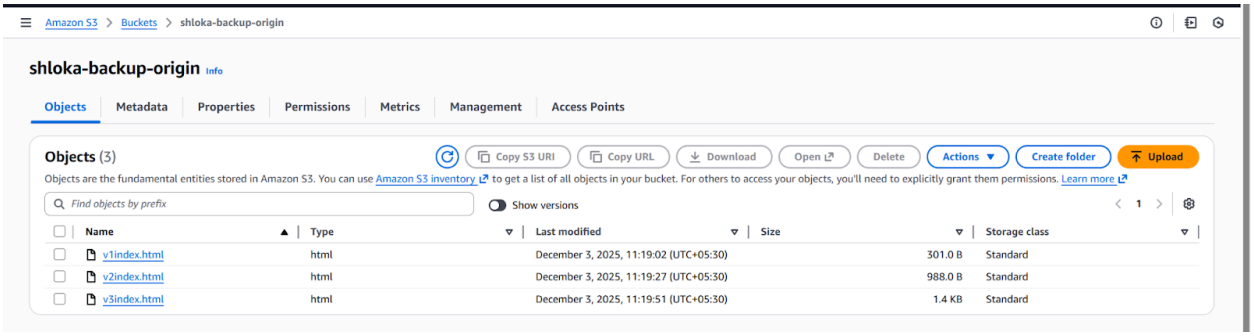
backup access patterns.



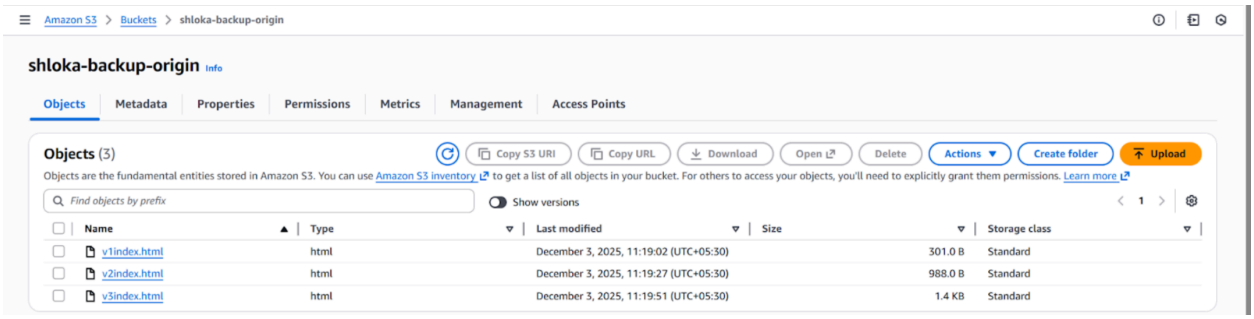
Step 5 : Uploading Objects

What was done:

- Uploaded 3 objects in the shloka-backup-origin



- While uploaded objects, added tags to each of them individually
  - v1index.html - env:prod
  - v2index.html - env:prod
  - v3index.html - env:test



Testing & Validation

Test 1: Replication Rule Functionality

Objective - Verify that only env=prod tagged objects replicate

Verdict - Passed. Only v1index.html and v2index.html were replicated