**Amazon S3: Custom Cost-Optimized Storage Strategy Implementation**

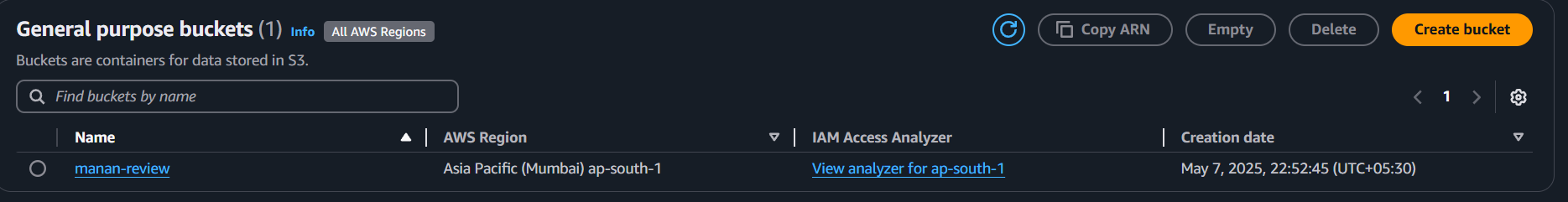
**Objective**

To implement a customized, cost-optimized Amazon S3 storage lifecycle for a dataset using manual tagging, intelligent tiering, lifecycle policies, and cost monitoring mechanisms via the AWS Console.

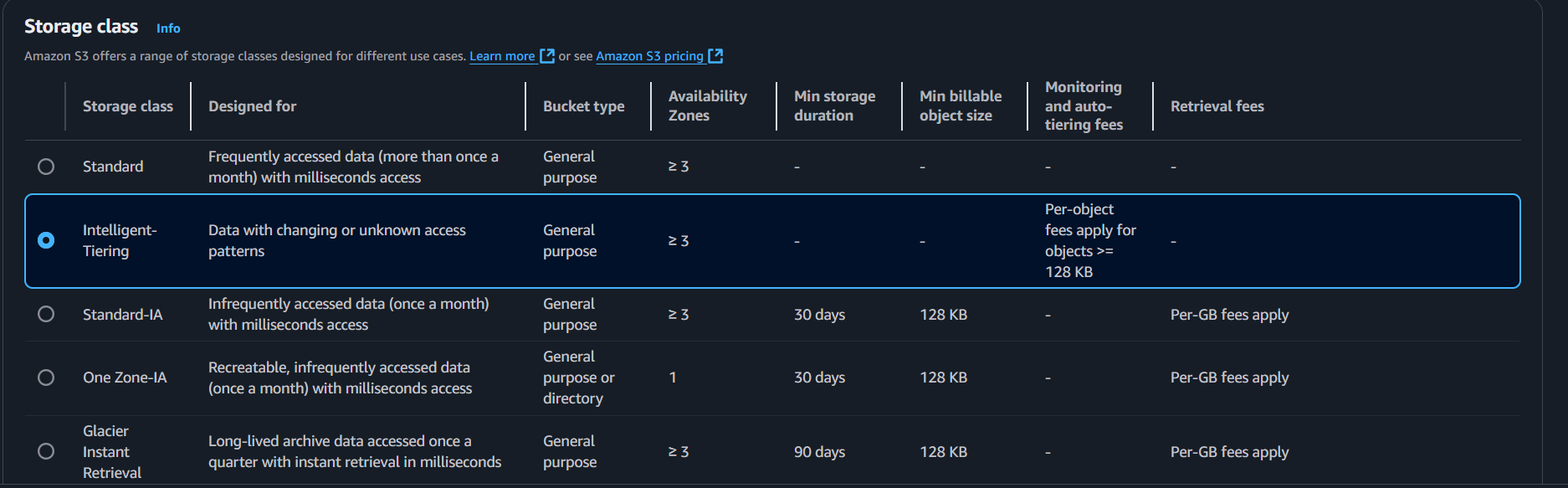
**Steps Completed**

**1. S3 Bucket Creation and Object Upload**

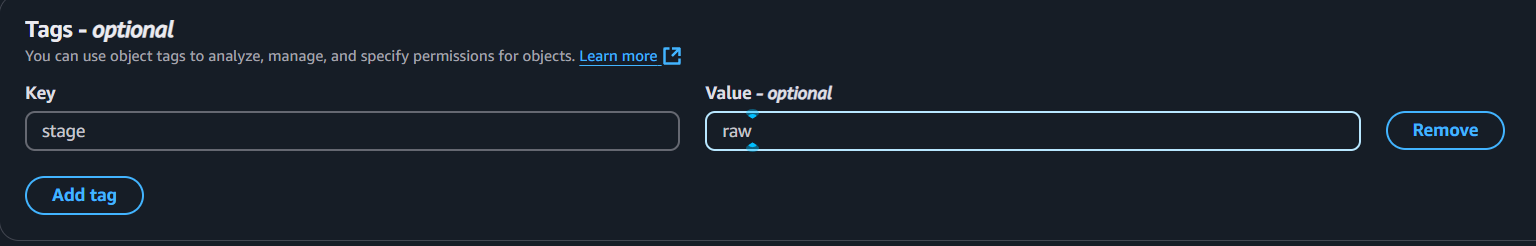
* **Bucket Name:** manan-reviews



* **Region:** ap-south-1
* **Versioning:** Disabled (can be enabled later for retention)
* **Encryption:** Enabled (SSE-S3)
* **Block Public Access:** All enabled (default and recommended)
* **Storage Class at Upload:** **Intelligent-Tiering**

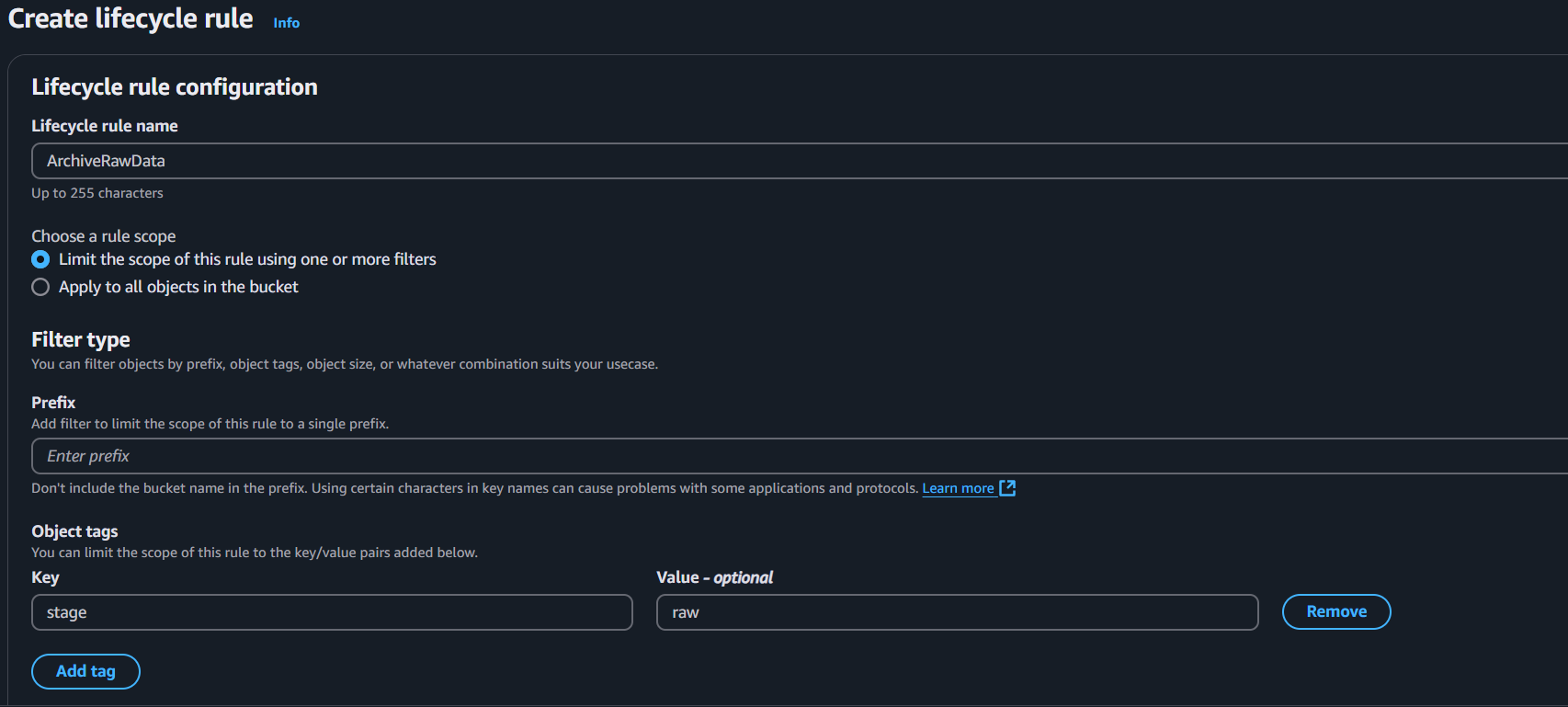


* **Tags Applied on Upload:**
  + Key: stage



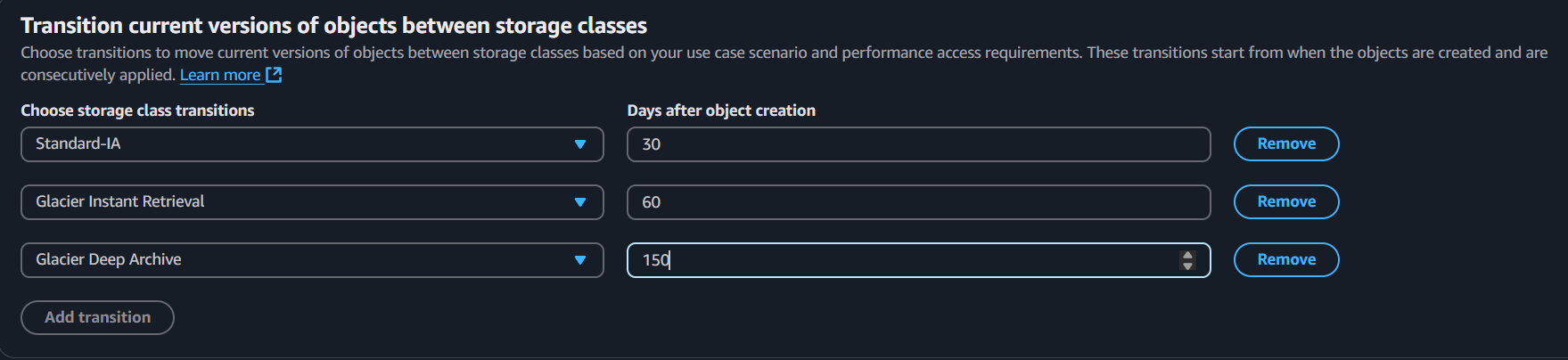
**2. Lifecycle Rule Configuration**

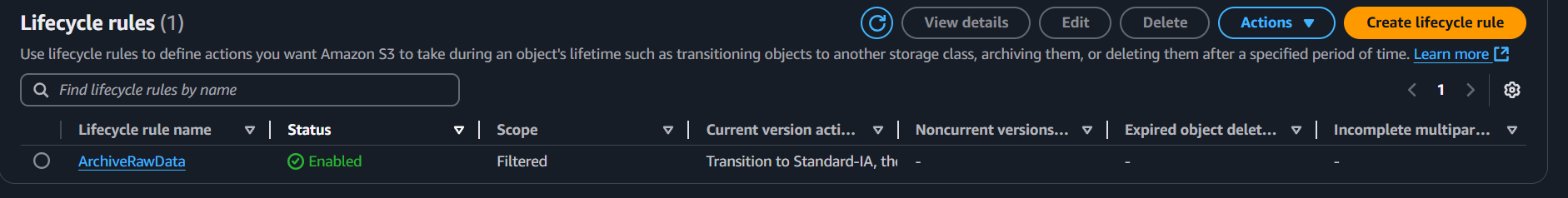
Separate lifecycle policies were created for each stage tag:



**Rule 1: Raw Data (stage=raw)**

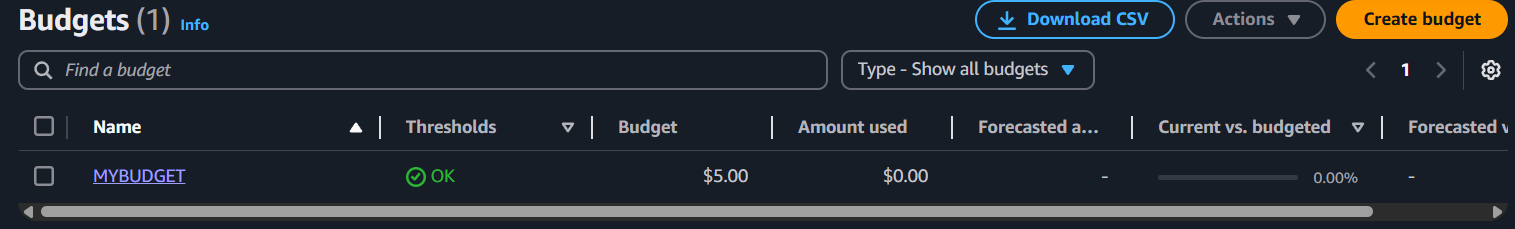
* Transition to **Standard-IA** after **30 days**
* Transition to **Glacier Instant Retrieval** after **60 days**
* Transition to **Glacier Deep Archive** after 150 **days**



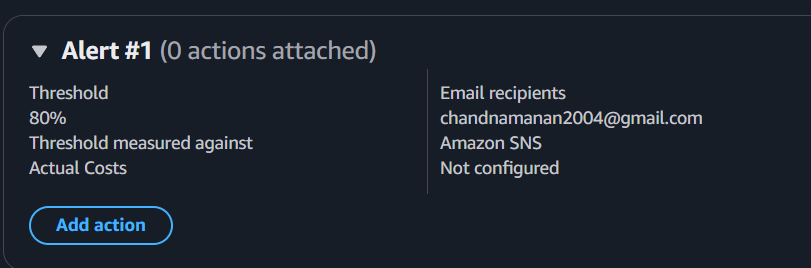


**NOTE:** Other rules can be created at later stages of the data such as stage=processed and stage=archive.

**3. Cost Budget Configuration**



* **Tool:** AWS Budgets (via Billing Console)
* **Budget Type:** Monthly cost budget
* **Service Scope:** Amazon S3
* **Alert Threshold:** 80% of budget limit
* **Notifications:** Email alerts configured



**Summary**

* **Bucket & Upload:** Created manan-reviews-data in ap-south-1 and uploaded objects tagged by stage (raw, processed, archive) using Intelligent-Tiering.
* **Lifecycle Policies:** Defined three tag-based rules to transition and expire data:
  + **Raw:** → Standard-IA (30 d) → Glacier Instant Retrieval (60 d) → Glacier Deep Archive (150 d)
* **Cost Monitoring:** Configured a monthly S3 budget with 80% alert threshold.

This strategy ensures automated, tag-driven data tiering and cost visibility without manual intervention.