# AWS Purging and Archival Strategies — S3, Athena, Redshift, Bedrock, DataZone, SageMaker

## Executive summary

This document defines purging and archival strategies for an AWS environment using Amazon S3, Amazon Athena, Amazon Redshift, Amazon Bedrock, Amazon DataZone, and Amazon SageMaker. It covers data lifecycle policies, automated archival, compliance requirements, retention timelines, cost optimization, and governance.

## Goals and principles

* **Regulatory compliance**: Retain data as long as required by legal and industry-specific mandates; securely purge data when retention period ends.
* **Cost efficiency**: Transition unused data from hot storage to cheaper archival tiers (S3 Glacier/Deep Archive).
* **Security**: Encrypt archived data with AWS KMS; use access policies and S3 Object Lock to prevent premature deletion.
* **Automation**: Use S3 Lifecycle rules, Glue jobs, and event-driven Lambda functions for automated purging and archival.
* **Discoverability**: Maintain metadata catalogs (Glue, DataZone) so archived datasets remain queryable and discoverable.
* **Separation of retention classes**: Define hot (active), warm (semi-active), and cold (archival) tiers.

## Common architectural pattern

1. **Data classification**: Classify datasets by type (logs, training data, query results, model artifacts).
2. **Lifecycle definition**: Define hot/warm/cold retention periods depending on service.
3. **Archival mechanism**: Use S3 Lifecycle policies, Redshift snapshot schedules, and Athena query result retention to move data.
4. **Purging mechanism**: Automated deletion after retention expiration, subject to compliance controls.
5. **Audit & governance**: Track purging and archival actions in CloudTrail; review quarterly.

## Per-service strategies

### Amazon S3 (primary storage & log lake)

* **Archival**:
  + Use S3 Lifecycle rules to transition objects from Standard -> Standard-IA -> Glacier Flexible Retrieval -> Deep Archive.
  + Apply storage class intelligently (e.g., frequently queried logs in Standard-IA; historical backups in Deep Archive).
* **Purging**:
  + Automatically delete objects after compliance-defined retention period.
  + Use S3 Object Expiration in Lifecycle rules.
* **Governance**:
  + Use Object Lock for WORM (write once, read many) requirements.
  + Encrypt all objects with SSE-KMS.

### Amazon Athena (query logs, results)

* **Archival**:
  + Store query results in S3; use lifecycle rules to transition to IA/Glacier after 90–180 days.
* **Purging**:
  + Automatically delete old query results after 1–2 years (or as per compliance).
  + Catalog metadata in Glue retained longer for audit purposes.
* **Governance**:
  + Enforce query result bucket retention with lifecycle rules.

### Amazon Redshift (cluster/serverless data)

* **Archival**:
  + Automate **snapshot schedules** to S3; configure cross-region snapshot copy for DR.
  + Use snapshot lifecycle to move older snapshots to Glacier.
* **Purging**:
  + Expire snapshots after defined retention (e.g., 30 days for dev, 7 years for prod compliance).
  + Use VACUUM and UNLOAD strategies to remove old staging tables and intermediate data.
* **Governance**:
  + Encrypt snapshots with KMS keys.
  + Log snapshot copy and deletion via CloudTrail.

### Amazon Bedrock (model usage, prompts)

* **Archival**:
  + Store metadata (model IDs, invocation metrics, latency, token counts) in S3 for long-term analysis.
  + Archive to Glacier/Deep Archive after 90 days.
* **Purging**:
  + Securely delete sensitive prompts/outputs after business-defined retention (e.g., 30–90 days).
  + Apply redaction pipelines before archival.
* **Governance**:
  + Prohibit logging raw user inputs unless allowed by compliance.

### Amazon DataZone (metadata catalog)

* **Archival**:
  + Metadata snapshots (catalog state, lineage) exported to S3; lifecycle management applies.
  + Archive lineage records to Glacier after 1 year.
* **Purging**:
  + Remove outdated catalogs/projects after 3–5 years, aligned with org data governance.
  + Automate cleanup with DataZone APIs + Lambda.
* **Governance**:
  + Maintain versioned metadata exports for recovery.
  + Encrypt and restrict metadata archives.

### Amazon SageMaker (training, inference, models)

* **Archival**:
  + Training datasets and outputs stored in S3 -> move to Glacier after project completion.
  + Model artifacts archived post-deployment but retained for reproducibility.
  + Endpoint invocation logs retained for 90 days in CloudWatch -> move to S3 -> Glacier.
* **Purging**:
  + Delete stale training artifacts, model versions not used in production.
  + Remove notebook instance snapshots older than compliance period.
* **Governance**:
  + Track deletions via CloudTrail.
  + Encrypt model artifacts and archived logs with KMS.

## Retention & lifecycle table (example)

| Service | Data Type | Hot retention | Warm retention | Cold/Archive retention | Purge |
| --- | --- | --- | --- | --- | --- |
| S3 | Logs, objects | 90 days | 1 year (IA) | 7 years (Glacier) | Delete after 7 yrs |
| Athena | Query results | 90 days | 1 year (IA) | N/A | Delete after 1 yr |
| Redshift | Snapshots, staging tables | 30 days | 1 year | 7 years (Glacier) | Delete after 7 yrs |
| Bedrock | Usage metadata | 30 days | 180 days | 2 years (Glacier) | Delete after 2 yrs |
| DataZone | Metadata, lineage | 1 year | 3 years | 5 years (Glacier) | Delete after 5 yrs |
| SageMaker | Models, training outputs | 180 days | 1 year | 3–5 years (Glacier) | Delete after 5 yrs |

## Security and compliance controls

* Encrypt all archives with KMS.
* Enable S3 Object Lock where WORM compliance is required.
* Track archival/purge events with CloudTrail.
* Review lifecycle rules quarterly for gaps.
* Document and approve retention schedules per compliance team.
* Validate deletion using verification jobs.

## Implementation roadmap

1. **Phase 1 (0–1 month)**
   * Define retention policies by service/data type.
   * Enable lifecycle rules on S3 buckets.
   * Configure Redshift snapshot schedules.
2. **Phase 2 (1–3 months)**
   * Automate archival pipelines with Lambda + EventBridge.
   * Configure Bedrock & SageMaker logging buckets with lifecycle.
   * Export DataZone metadata to S3.
3. **Phase 3 (3–6 months)**
   * Enable Object Lock for compliance-critical buckets.
   * Implement automated purge verification scripts.
   * Review with compliance and governance boards.

*Prepared for: AWS Environment — S3, Athena, Redshift, Bedrock, DataZone, SageMaker*

*Document version: 1.0 — created on 2025-09-12*