

Enterprise Centralized Logging and Observability Architecture on AWS

Author: Adedayo

Specialization: Cloud Security & Observability

Platform: Amazon Web Services (AWS)

1. Problem Statement

An organization operating multiple AWS accounts lacked centralized visibility into logs and security findings. Logs were dispersed across accounts, limiting detection capabilities. A unified logging and SIEM integration architecture was required.

I implemented this solution to ensure that critical logs from distributed workloads are consistently collected, securely stored, retained according to compliance requirements, and analyzed through a centralized SIEM platform.

2. Objectives

The primary objectives of this implementation were to:

- Centralize AWS service and application logs
- Improve security monitoring and incident investigation
- Support regulatory and compliance requirements
- Enable real-time detection and alerting
- Ensure secure and durable log storage
- Reduce operational blind spots

3. Logging Architecture Overview

The centralized logging platform was built using:

- AWS CloudTrail
- Amazon CloudWatch Logs
- Amazon Kinesis Data Firehose
- AWS Lambda
- Amazon OpenSearch
- Amazon S3

All service and application logs are routed through CloudWatch and streamed into a secure analytics platform.

4. CloudTrail Configuration

CloudTrail was configured as a multi-region, organization-level trail.

The screenshot shows the AWS CloudTrail Trails page. At the top, there is a green banner with the text "Trail successfully created". Below the banner, the page title is "Trails". There is a "Create trail" button. The main table lists one trail:

Name	Home region	Multi-region trail	ARN	Insights	Organization trail	S3 bucket	Log file prefix	CloudWatch Logs log group	Status
CloudsecTrail	US East (N. Virginia)	Yes	arn:aws:cloudtrail:us-east-1:879381257906:trail/CloudsecTrail	Enabled	Yes	aws-cloudtrail-logs-879381257906-dd20b7e0	-	arn:aws:logs:us-east-1:879381257906:log-group:aws-cloudtrail-logs-879381257906-9ecf6266-*	Logging

The configuration includes:

- Management event logging
- CloudTrail Insights for anomaly detection
- Centralized S3 storage
- Streaming to CloudWatch Logs

This ensured full visibility into account activity across regions.

5. Log Retention and Lifecycle Management

CloudWatch Log Groups were configured with defined retention periods to prevent indefinite storage.

CloudTrail logs stored in S3 are managed using lifecycle policies that:

- Retain logs in S3 Standard for 90 days

The screenshot shows the AWS CloudWatch Log Management interface. On the left, there's a navigation sidebar with sections like 'CloudWatch' (selected), 'Favorites and recents', 'Logs' (selected), and 'Metrics'. Under 'Logs', there are sub-options like 'Log Management' (selected), 'Log Anomalies', 'Live Tail', 'Logs Insights', and 'Contributor Insights'. The main content area is titled 'Log groups (1)'. It shows a single log group named 'f6266' with a 'Standard' type, 'Configure' button, and 'Off' status for deletion. A search bar at the top says 'Filter log groups or try pattern search' with 'Exact match' checked. Below the search bar are various filter options: Log class, Anomaly d..., Deletion..., Data ..., Sensit..., Retenti..., Metric ..., Contribut..., and Subscri... with dropdown arrows. A time range selector shows '3 months'. At the top right are buttons for 'Actions', 'View in Logs Insights', 'Start tailing', and 'Create log group'.

- Archive older logs to S3 Glacier

- Delete logs after one year

The screenshot shows the Amazon S3 Lifecycle rule configuration page. The URL is 'Amazon S3 > Buckets > aws-cloudtrail-logs-879581257906-dd20b7e0 > Lifecycle configuration > cloudseclogretentioncycle'. The configuration details are as follows:

- Lifecycle rule name:** cloudseclogretentioncycle
- Status:** Enabled
- Scope:** Entire bucket
- Prefix:** -
- Object tags:** -
- Minimum object size:** -
- Maximum object size:** -

Review transition and expiration actions:

	Current version actions	Noncurrent versions actions
Day 0	<ul style="list-style-type: none"> Objects uploaded 	Day 0 No actions defined.
Day 90	<ul style="list-style-type: none"> Objects move to Glacier Instant Retrieval 	
Day 365	<ul style="list-style-type: none"> Objects expire 	

This approach aligns with common compliance standards such as PCI-DSS while controlling storage costs.

6. Centralized SIEM Platform (Amazon OpenSearch)

Amazon OpenSearch was deployed as the centralized log analytics and SIEM platform.

The screenshot shows the Amazon OpenSearch Service console with the 'cloudssecdomain' domain selected. The 'Cluster configuration' tab is active. Key details shown include:

- Name:** cloudssecdomain
- Domain ARN:** arn:aws:es:us-east-1:879381257906:domain/cloudssecdomain
- Deployment option(s):** 3-AZ with standby
- Domain processing status:** Active
- Configuration change status:** Completed
- Cluster health:** Green
- Version Info:** OpenSearch 3.3 (latest)
- Service software version:** OpenSearch_3_3_R20251121-P3 (latest)
- OpenSearch Dashboards URL (IPv4):** https://search-cloudssecdomain-ueat4lzgysjf2pqmbyxoc72rm.us-east-1.es.amazonaws.com/_dashboards
- Domain endpoint:** https://search-cloudssecdomain-ueat4lzgysjf2pqmbyxoc72rm.us-east-1.es.amazonaws.com

The domain was configured with:

- No public internet exposure
- VPC-based deployment
- Fine-grained access control
- IAM-based authentication
- Encrypted storage
- Node-to-node encryption

Access was restricted using security groups to authorized ingestion services.

7. Secure Log Ingestion Pipeline

7.1 CloudWatch Subscription Filters

Subscription filters were created to forward all CloudTrail events to Kinesis Data Firehose.

No filtering was applied to ensure complete audit coverage.

7.2 Kinesis Data Firehose Configuration

Firehose was configured with:

- OpenSearch as primary destination

- S3 backup destination
- VPC delivery enabled

Amazon Data Firehose > Firehose streams > PUT-OPS-KKey6

PUT-OPS-KKey6 Info

Firehose stream details

Status Active	Destination Amazon OpenSearch Service	Data transformation Enabled
Source Direct PUT	ARN arn:aws:firehose:us-east-1:879381257906:deliverystream/PUT-OPS-KKey6	

Test with demo data Info
Ingest simulated data to test the configuration of your Firehose stream. Standard Amazon Data Firehose charges apply.

Monitoring **Configuration** **Destination error logs** **Backup error logs**

Firehose stream metrics Info

Investigate with AI - new 3h 1d 1w UTC timezone C

Incoming bytes Bytes	Incoming put requests Count	Incoming records Count
-------------------------	--------------------------------	---------------------------

An IAM role was created with permissions for:

- Writing to OpenSearch
- Delivering data to S3
- Managing network interfaces

7.3 Lambda Log Transformation

A Lambda function was integrated into Firehose to perform log transformation.

The screenshot shows the AWS Lambda function editor interface. In the top navigation bar, it says "Lambda > Functions > firehoseLambda". The left sidebar has sections for "EXPLORER", "FIREHOSELAMBDA" (which contains __pycache__ and lambda_function.py), "TEST EVENTS (NONE SELECTED)", and "+ Create new test event". The main area is titled "lambda_function.py" and contains the following Python code:

```
1 import base64
2 import gzip
3 import json
4
5 def lambda_handler(event, context):
6     output = []
7
8     for record in event['records']:
9         data = base64.b64decode(record['data'])
10
11         # CloudWatch subscription data is often gzipped
12         try:
13             data = gzip.decompress(data)
14         except Exception:
15             pass
16
17         try:
18             text = data.decode('utf-8')
19         except Exception:
20             output.append({
21                 'recordId': record['recordId'],
22                 'result': 'ProcessingFailed',
23                 'data': record['data']
24             })
25         continue
26
27         # Output must be UTF-8 JSON line(s)
28         try:
29             obj = json.loads(text)
```

Functions included:

- Normalizing log formats
- Adding metadata
- Ensuring indexing compatibility

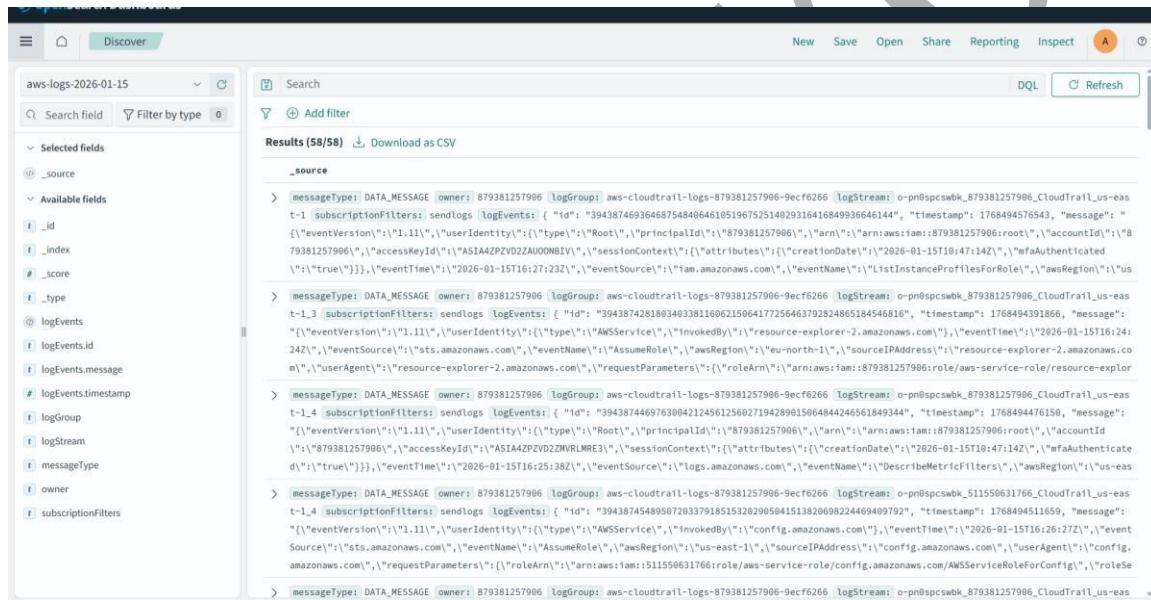
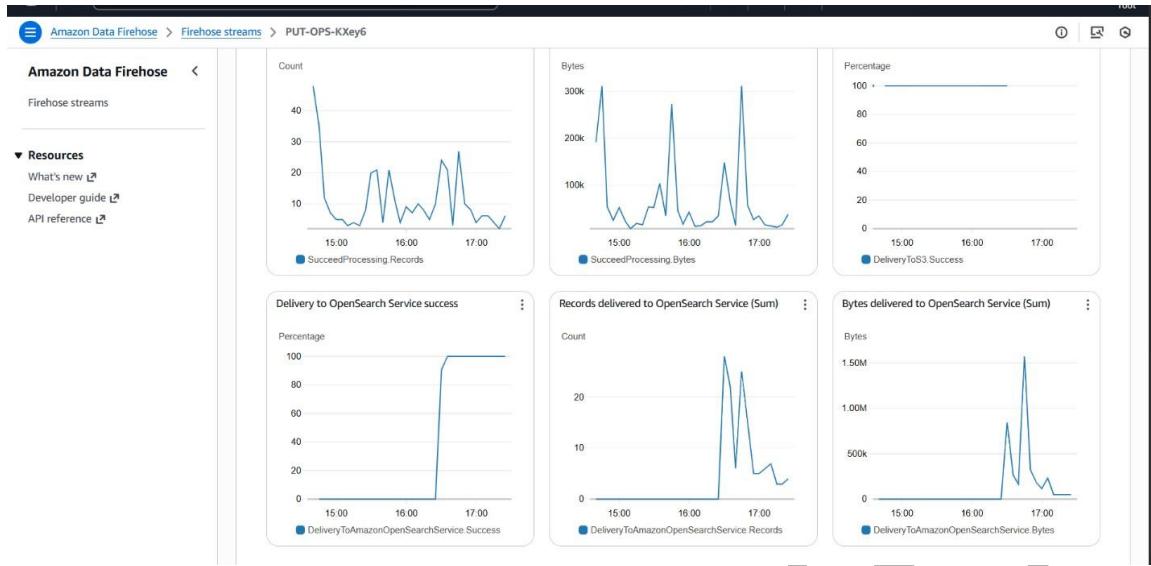
This improved searchability and consistency.

8. Pipeline Validation and Testing

The logging pipeline was validated end-to-end.

After resolving VPC network restrictions, log ingestion was confirmed in OpenSearch.

Indexed records were verified to be searchable and complete.



9. Workload Log Integration

9.1 AWS Lambda Logging

Lambda workloads were verified to emit logs to CloudWatch.

Test invocations confirmed correct log group and stream creation.

CloudWatch > Log management > /aws/lambda/firehoseLambda > 2026/01/15/[SLATEST]01b86768a39d44e3b06e29c27f82419e

Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

Filter events - press enter to search

Actions ▾ Start tailing Create metric filter

Clear 1m 30m 1h 12h Custom UTC timezone

Display ▾

Timestamp	Message
2026-01-15T18:20:37.669Z	END RequestId: bac6f639-8fa5-49d1-8c9c-d30d849c8aab
2026-01-15T18:20:37.669Z	REPORT RequestId: bac6f639-8fa5-49d1-8c9c-d30d849c8aab Duration: 38.07 ms Billed Duration: 39 ms Memory Size: 128 MB Max Memory Used: 128 MB
2026-01-15T18:22:48.077Z	START RequestId: 176329b2-b0c3-4428-ab46-d821c67c2a2a Version: \$LATEST
2026-01-15T18:22:48.130Z	END RequestId: 176329b2-b0c3-4428-ab46-d821c67c2a2a
2026-01-15T18:22:48.130Z	REPORT RequestId: 176329b2-b0c3-4428-ab46-d821c67c2a2a Duration: 52.01 ms Billed Duration: 53 ms Memory Size: 128 MB Max Memory Used: 128 MB
2026-01-15T18:23:56.161Z	START RequestId: fcf351bb-c644-4e67-b81f-13a3a53038e0 Version: \$LATEST
2026-01-15T18:23:56.194Z	END RequestId: fcf351bb-c644-4e67-b81f-13a3a53038e0
2026-01-15T18:23:56.194Z	REPORT RequestId: fcf351bb-c644-4e67-b81f-13a3a53038e0 Duration: 33.01 ms Billed Duration: 34 ms Memory Size: 128 MB Max Memory Used: 128 MB
2026-01-15T18:23:56.194Z	REPORT RequestId: fcf351bb-c644-4e67-b81f-13a3a53038e0 Duration: 33.01 ms Billed Duration: 34 ms Memory Size: 128 MB Max Memory Used: 45 MB

No newer events at this moment. Auto-archived. [Resume](#)

Back to top ▾

9.2 ECS Container Logging

ECS Fargate workloads were configured using the awslogs driver.

Application logs were verified in dedicated CloudWatch log groups.

This ensured container workloads were centrally monitored.

CloudWatch > Log management > /ecs/cloudses > ecs/test-logging-container/d9bf630042c94fdb9338b1f2bc792222

Log events

You can use the filter bar below to search for and match terms, phrases, or values in your log events. [Learn more about filter patterns](#)

Filter events - press enter to search

Actions ▾ Start tailing Create metric filter

Clear 1m 30m 1h 12h Custom UTC timezone

Display ▾

Timestamp	Message
2026-01-15T19:05:40.234Z	10-listen-on-ipv4-by-default.sh: info: listening on the checksum of /etc/nginx/conf.d/default.conf
2026-01-15T19:05:40.221Z	10-listen-on-ipv6-by-default.sh: info: Enabled listen on IPv6 in /etc/nginx/conf.d/default.conf
2026-01-15T19:05:40.222Z	/docker-entrypoint.sh: Sourcing /docker-entrypoint.d/15-local-resolvers.envsh
2026-01-15T19:05:40.222Z	/docker-entrypoint.sh: Launching /docker-entrypoint.d/20-envsubst-on-templates.sh
2026-01-15T19:05:40.234Z	/docker-entrypoint.sh: Launching /docker-entrypoint.d/30-tune-worker-processes.sh
2026-01-15T19:05:40.235Z	/docker-entrypoint.sh: Configuration complete; ready for start up
2026-01-15T19:05:40.239Z	2026/01/15 19:05:40 [notice] 1#1: using the "epoll" event method
2026-01-15T19:05:40.239Z	2026/01/15 19:05:40 [notice] 1#1: nginx/1.28.1
2026-01-15T19:05:40.239Z	2026/01/15 19:05:40 [notice] 1#1: built by gcc 14.2.0 (Debian 14.2.0-19)
2026-01-15T19:05:40.239Z	2026/01/15 19:05:40 [notice] 1#1: 05: Linux 5.10.245.983.amzn2.x86_64
2026-01-15T19:05:40.239Z	2026/01/15 19:05:40 [notice] 1#1: getrlimit(RLIMIT_NOFILE): 65535:65535
2026-01-15T19:05:40.240Z	2026/01/15 19:05:40 [notice] 1#1: start worker processes
2026-01-15T19:05:40.240Z	2026/01/15 19:05:40 [notice] 1#1: start worker process 28

10. Durability and Backup Strategy

All logs delivered through Firehose are backed up to S3.

Lifecycle policies archive data to Glacier for long-term retention.

This provides durability, disaster recovery, and compliance support.

11. Governance and Documentation

All logging configurations, access policies, and retention standards were documented.

This supported:

- Audit reviews
- Compliance reporting
- Incident investigations
- Operational continuity

12. Outcomes and Impact

This implementation delivered:

- Centralized security visibility
- Improved incident response capability
- Compliance-aligned log retention
- Secure SIEM deployment
- Reliable log delivery
- Reduced operational blind spots

13. Conclusion

I designed and implemented a secure, scalable, and compliant centralized logging and observability platform on AWS.

Through automated log collection, secure analytics, and structured retention policies, this solution supports enterprise security monitoring, investigations, and regulatory requirements.