Implementation Guide

Centralized Logging with OpenSearch

Centralized Logging with OpenSearch: Implementation Guide

Table of Contents

Build your own centralized log analytics platform with Amazon OpenSearch Service in 20 minutes

Publication date: *March 2023 (last update: March 2024)*

The Centralized Logging with OpenSearch solution provides comprehensive log management and analysis functions to help you simplify the build of log analytics pipelines. Built on top of Amazon OpenSearch Service, the solution allows you to streamline log ingestion, log processing, and log visualization. You can leverage the solution in multiple use cases such as to abide by security and compliance regulations, achieve refined business operations, and enhance IT troubleshooting and maintenance.

Use this navigation table to quickly find answers to these questions:

If you want to ...	Read...
Know the cost for running this solution	Cost
Understand the security considerations for this solution	Security
Know which AWS Regions are supported for this solution	Supported AWS Regions
Get started with the solution quickly to import an Amazon OpenSearch Service domain, build a log analytics pipeline, and access the built-in dashboard	Getting started
Learn the operations related to Amazon OpenSearch Service domains	Domain management
Walk through the processes of building log analytics pipelines	AWS Services logs and Application logs

This implementation guide describes architectural considerations and configuration steps for deploying the Centralized Logging with OpenSearch solution in the AWS cloud. It includes links to

CloudFormation templates that launches and configures the AWS services required to deploy this solution using AWS best practices for security and availability.

The guide is intended for IT architects, developers, DevOps, data engineers with practical experience architecting on the AWS Cloud.

Features and benefits

The solution has the following features:

All-in-one log ingestion:

provides a single web console to ingest both application logs and AWS service logs into log analytics engines. For supported AWS service logs, refer to AWS Service Logs. For supported application logs, refer to Application Logs.

Codeless log processor:

supports log processor plugins developed by AWS. You are allowed to enrich the raw log data through a few steps on the web console.

Out-of-the-box dashboard template:

offers a collection of reference designs of visualization templates, for both commonly used software such as Nginx and Apache HTTP Server, and AWS services such as Amazon S3 and AWS CloudTrail.

Use cases

The solution can be applied to the following use cases:

- **Security and compliance regulations**

  Comply with regulatory requirements such as MLPS, GDPR, PCI DSS, and HIPAA. Easily store equipment, network, and application logs in a centralized place for log auditing and threat detection.
- **Business operations and data analysis**

  Identify trends and patterns in minutes, and build interactive and intuitive visualization. Derive business insights from logs and empower business decisions with data.

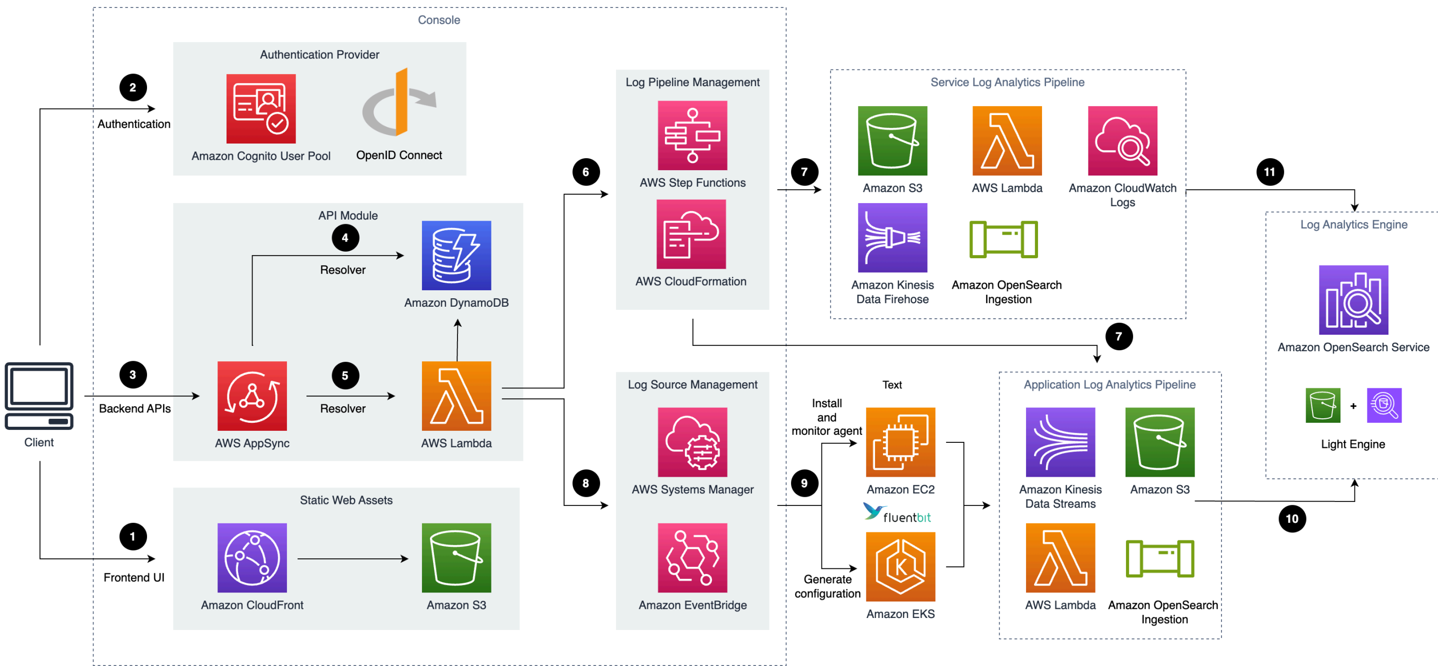
- **Application and infrastructure troubleshooting**

  Monitor both application and cloud infrastructure logs with ease, understand and resolve the root cause of issues quickly. Improve observability of your workloads, and achieve better business stability.

Architecture overview

Deploying this solution with the default parameters builds the following environment in the AWS Cloud.

Architecture diagram



Centralized Logging with OpenSearch architecture overview

This solution deploys the AWS CloudFormation template in your AWS Cloud account and completes the following settings.

1. Amazon CloudFront distributes the frontend web UI assets hosted in Amazon S3 bucket.
2. Amazon Cognito user pool or OpenID Connector (OIDC) can be used for authentication.
3. AWS AppSync provides the backend GraphQL APIs.
4. Amazon DynamoDB stores the solution related information as backend database.
5. AWS Lambda interacts with other AWS Services to process core logic of managing log pipelines or log agents, and obtains information updated in DynamoDB tables.
6. AWS Step Functions orchestrates on-demand AWS CloudFormation deployment of a set of predefined stacks for log pipeline management. The log pipeline stacks deploy separate AWS

resources and are used to collect and process logs and ingest them into Amazon OpenSearch Service for further analysis and visualization.

7. Service Log Pipeline or Application Log Pipeline are provisioned on demand via Centralized Logging with OpenSearch console.
8. AWS Systems Manager and Amazon EventBridge manage log agents for collecting logs from application servers, such as installing log agents (Fluent Bit) for application servers and monitoring the health status of the agents.
9. Amazon EC2 or Amazon EKS installs Fluent Bit agents, and uploads log data to application log pipeline.
10. Application log pipelines read, parse, process application logs and ingest them into Amazon OpenSearch Service domains or Light Engine.
11. Service log pipelines read, parse, process AWS service logs and ingest them into Amazon OpenSearch Service domains or Light Engine.

After deploying the solution, you can use AWS WAF to protect CloudFront or AppSync. Moreover, you can follow this guide to configure your WAF settings to prevent GraphQL schema introspection.

This solution supports two types of log pipelines: **Service Log Analytics Pipeline** and **Application Log Analytics Pipeline**.

Service log analytics pipeline

Centralized Logging with OpenSearch supports log analysis for AWS services, such as Amazon S3 access logs, and Elastic Load Balancing access logs. For a complete list of supported AWS services, refer to Supported AWS Services.

This solution ingests different AWS service logs using different workflows.

> **Note**
>
> Centralized Logging with OpenSearch supports cross-account log ingestion. If you want to ingest the logs from another AWS account, the resources in the **Sources** group in the architecture diagram will be in another account.

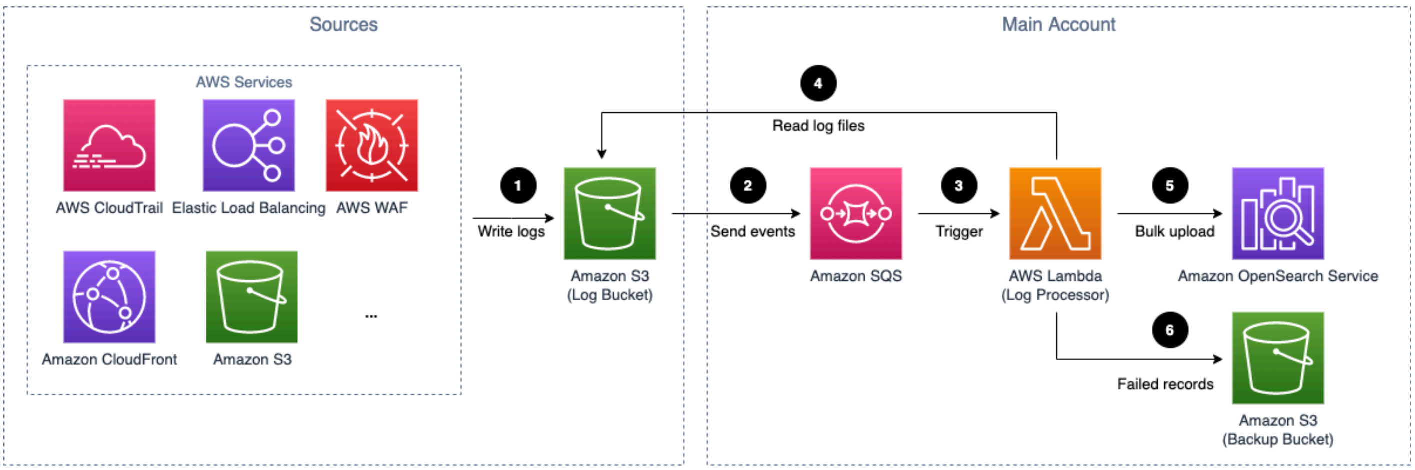
Logs through Amazon S3

This section is applicable to Amazon S3 access logs, CloudFront standard logs, CloudTrail logs (S3), Elastic Load Balancing access logs, WAF logs, VPC Flow logs (S3), AWS Config logs, Amazon RDS/ Aurora logs, and AWS Lambda Logs.

The workflow supports the following scenarios:

- **Logs to Amazon S3 directly (Amazon OpenSearch Service for log analytics)**

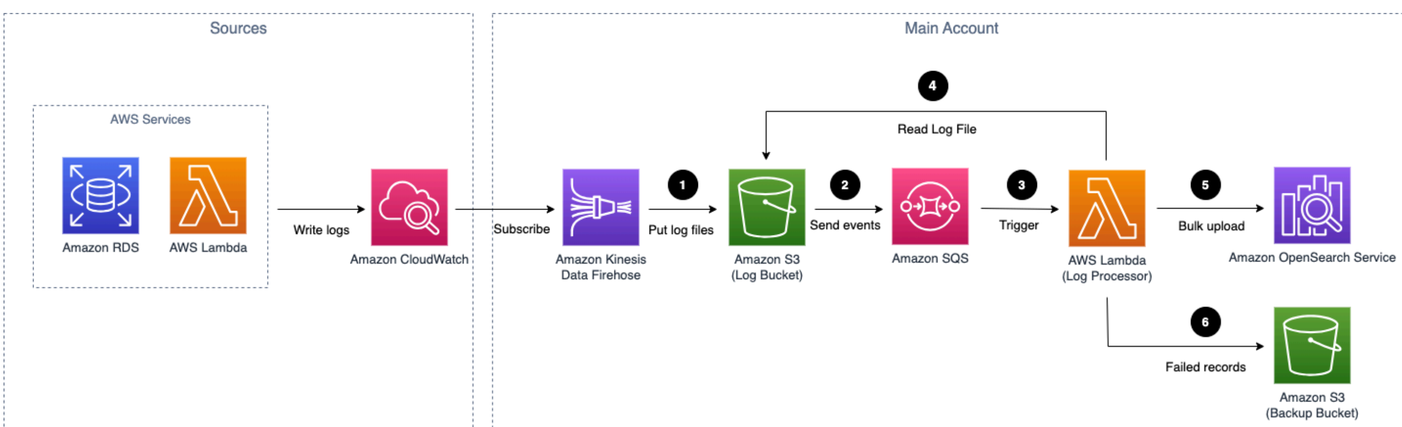
  In this scenario, the service directly sends logs to Amazon S3.



Amazon S3 based service log pipeline architecture

- **Logs to Amazon S3 via Firehose (Amazon OpenSearch Service for log analytics)**

  In this scenario, the service cannot directly put their logs to Amazon S3. The logs are sent to Amazon CloudWatch, and Firehose is used to subscribe the logs from CloudWatch Log Group and then put logs into Amazon S3.



Amazon S3 (via Kinesis Data Firehose) based service log pipeline architecture

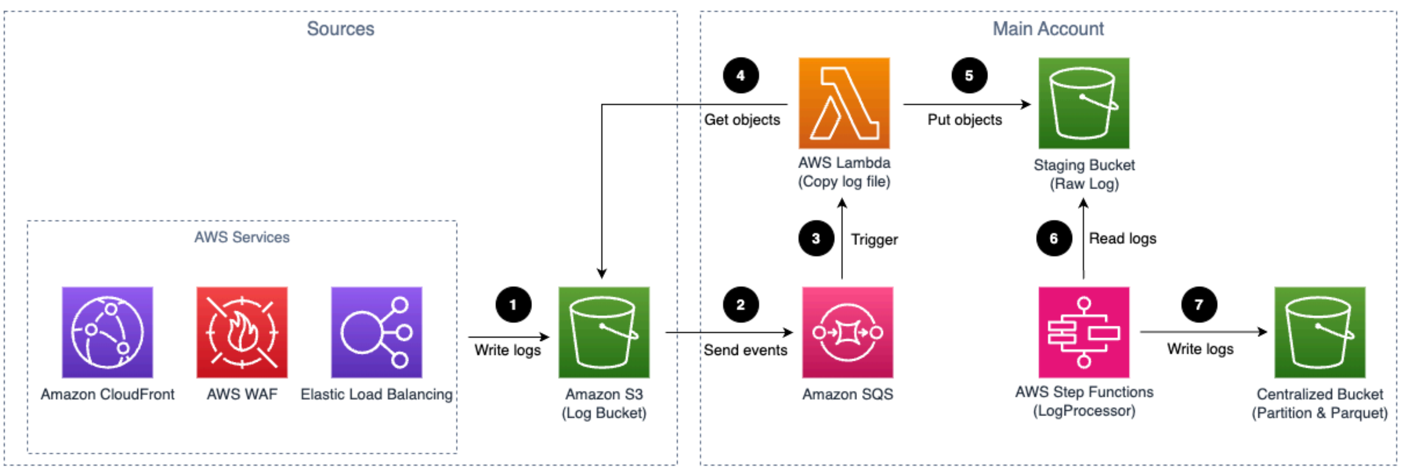
The log pipeline runs the following workflow:

1. AWS service logs are stored in an Amazon S3 bucket (Log Bucket).
2. An event notification is sent to Amazon SQS using S3 Event Notifications when a new log file is created.
3. Amazon SQS initiates the log processor Lambda function to run.
4. The log processor reads and processes the log files.
5. The log processor ingests the logs into the Amazon OpenSearch Service.
6. Logs that fail to be processed are exported to Amazon S3 bucket (Backup Bucket).

For cross-account ingestion, the AWS Services store logs in Amazon S3 bucket in the member account, and other resources remain in central logging account.

- **Logs to Amazon S3 directly (Light Engine for log analytics)**

In this scenario, the service directly sends logs to Amazon S3.



Amazon S3 (via Kinesis Data Firehose) based service log pipeline architecture

The log pipeline runs the following workflow:

1. AWS service logs are stored in an Amazon S3 bucket (Log Bucket).
2. An event notification is sent to Amazon SQS using S3 Event Notifications when a new log file is created.
3. Amazon SQS initiates AWS Lambda.

4. AWS Lambda gets objects from the Amazon S3 log bucket.
5. AWS Lambda puts objects to the staging bucket.
6. The log processor, AWS Step Functions, processes raw log files stored in the staging bucket in batches.
7. The log processor, AWS Step Functions, converts log data into Apache Parquet format and automatically partitions all incoming data based on criteria including time and region.

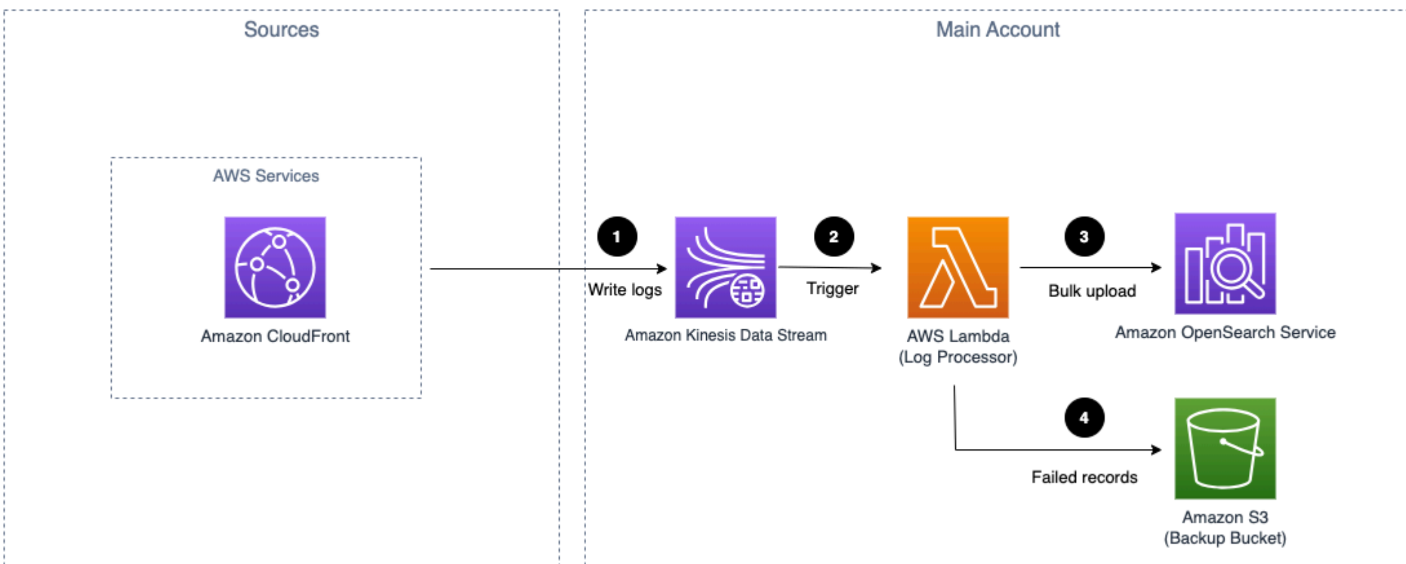
Logs through Amazon Kinesis Data Streams

This section is applicable to CloudFront real-time logs, CloudTrail logs (CloudWatch), and VPC Flow logs (CloudWatch).

The workflow supports two scenarios:

- **Logs to KDS directly**

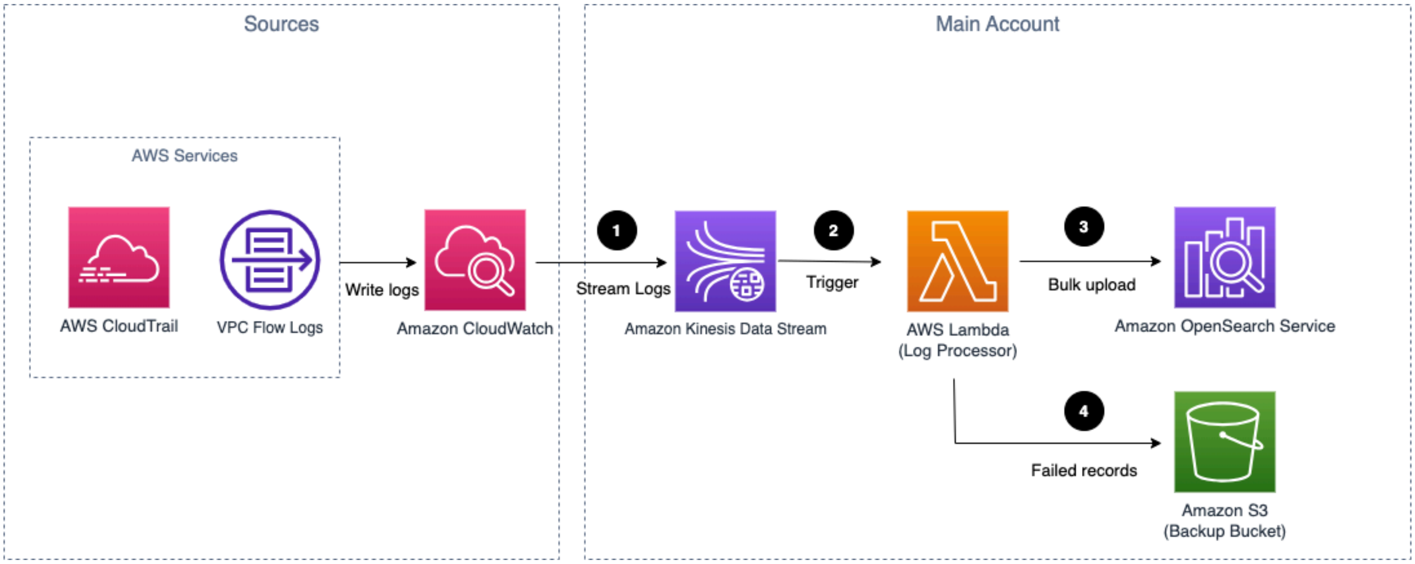
  In this scenario, the service directly streams logs to Amazon Kinesis Data Streams.



Amazon Kinesis Data Streams based service log pipeline architecture

- **Logs to KDS via subscription**

  In this scenario, the service delivers the logs to CloudWatch Log Group, and then CloudWatch Logs stream the logs in real-time to KDS as the subscription destination.



Amazon Kinesis Data Streams (via subscription) based service log pipeline architecture

The log pipeline runs the following workflow:

1. AWS Services logs are streamed to Kinesis Data Stream.
2. KDS initiates the log processor Lambda function to run.
3. The log processor processes and ingests the logs into the Amazon OpenSearch Service.
4. Logs that fail to be processed are exported to Amazon S3 bucket (Backup Bucket).

For cross-account ingestion, the AWS Services store logs on Amazon CloudWatch log group in the member account, and other resources remain in central logging account.

Warning

This solution does not support cross-account ingestion for CloudFront real-time logs.

Application log analytics pipeline

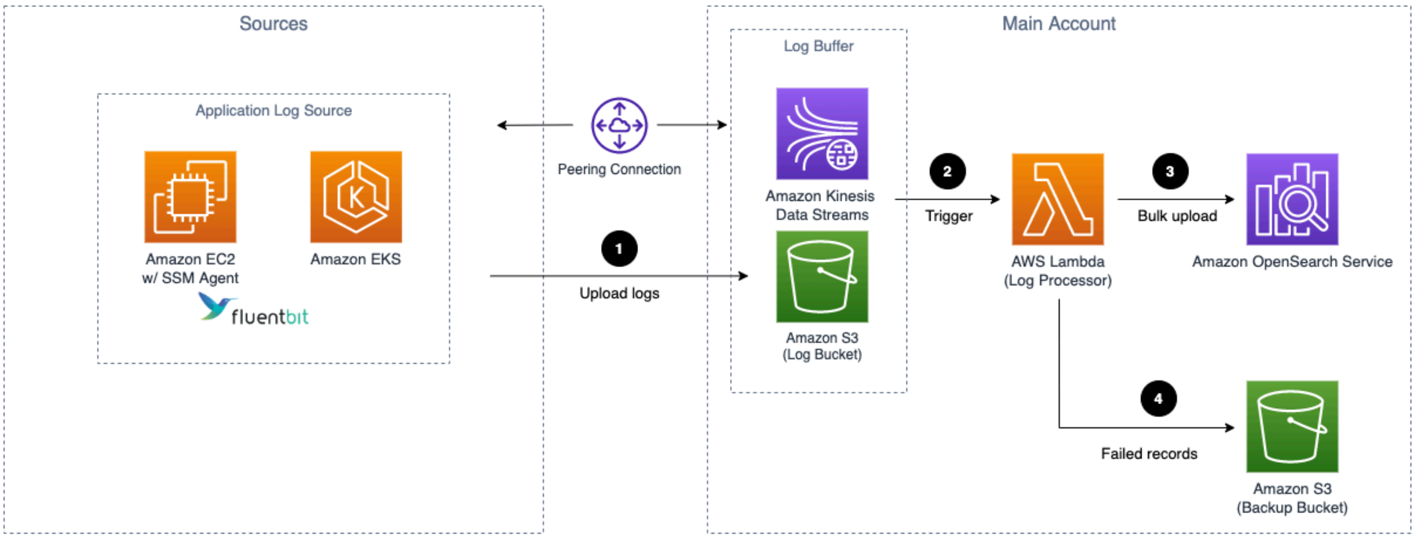
Centralized Logging with OpenSearch supports log analysis for application logs, such as Nginx/ Apache HTTP Server logs or custom application logs.

Note

Centralized Logging with OpenSearch supports [cross-account log ingestion](). If you want to ingest logs from the same account, the resources in the **Sources** group will be in the same account as your Centralized Logging with OpenSearch account. Otherwise, they will be in another AWS account.

Logs from Amazon EC2/Amazon EKS

- **Logs from Amazon EC2/Amazon EKS (Amazon OpenSearch Service for log analytics)**

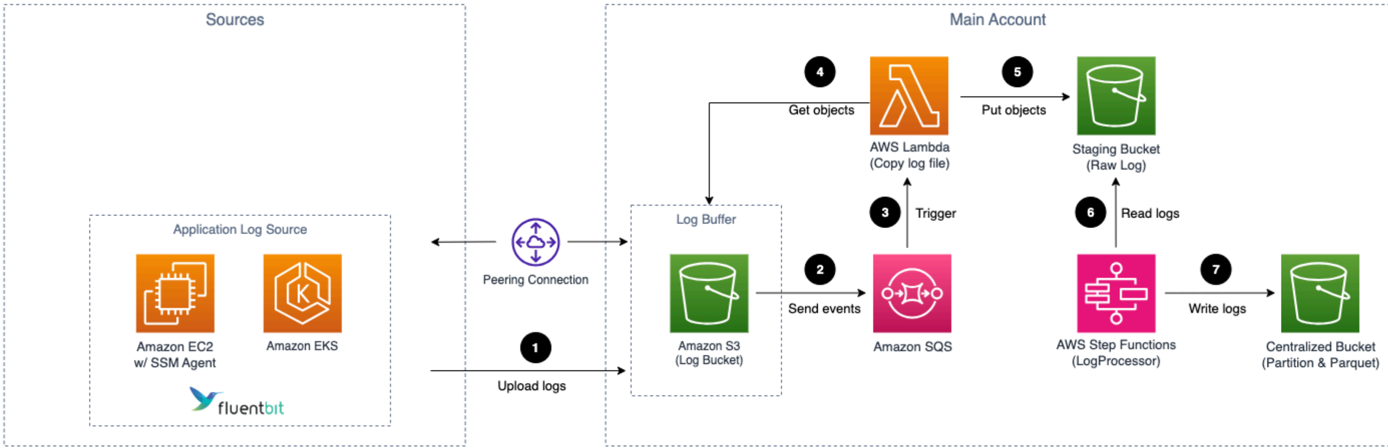


Application log pipeline architecture for EC2/EKS

The log pipeline runs the following workflow:

1. Fluent Bit works as the underlying log agent to collect logs from application servers and send them to an optional Log Buffer, or ingest into OpenSearch domain directly.
2. The Log Buffer triggers the Lambda function (log processor) to run.
3. The log processor reads and processes the log records and ingests the logs into the OpenSearch domain.
4. Logs that fail to be processed are exported to an Amazon S3 bucket (Backup Bucket).

- **Logs from Amazon EC2/Amazon EKS (Light Engine for log analytics)**



Application log pipeline architecture for EC2/EKS

The log pipeline runs the following workflow:

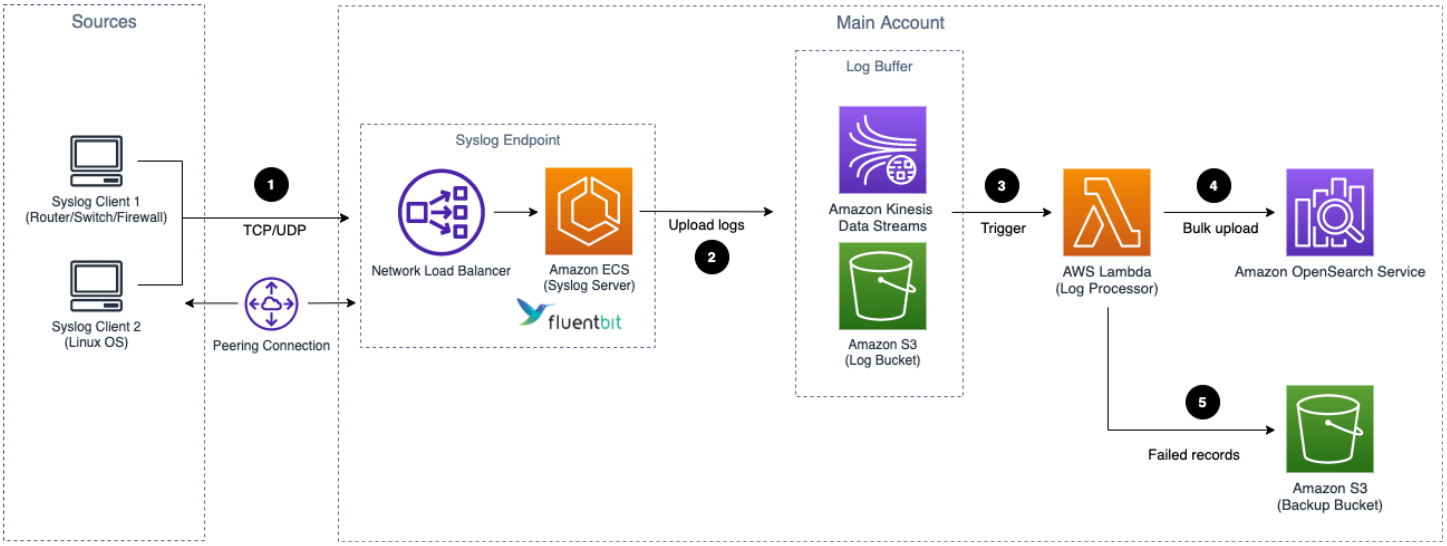
1. Fluent Bit works as the underlying log agent to collect logs from application servers and send them to an optional Log Buffer.
2. An event notification is sent to Amazon SQS using S3 Event Notifications when a new log file is created.
3. Amazon SQS initiates AWS Lambda.
4. AWS Lambda gets objects from the Amazon S3 log bucket.
5. AWS Lambda puts objects to the staging bucket.
6. The log processor, AWS Step Functions, processes raw log files stored in the staging bucket in batches.
7. The log processor, AWS Step Functions, converts log data into Apache Parquet format and automatically partitions all incoming data based on criteria including time and region .

Logs from Syslog Client

Important

1. Make sure your Syslog generator/sender's subnet is connected to Centralized Logging with OpenSearch' **two** private subnets. You need to use VPC Peering Connection or Transit Gateway to connect these VPCs.

2. The NLB together with the ECS containers in the architecture diagram will be provisioned only when you create a Syslog ingestion and be automated deleted when there is no Syslog ingestion.



Application log pipeline architecture for Syslog

1. Syslog client (like Rsyslog) send logs to a Network Load Balancer (NLB) in Centralized Logging with OpenSearch's private subnets, and NLB routes to the ECS containers running Syslog servers.
2. Fluent Bit works as the underlying log agent in the ECS Service to parse logs, and send them to an optional Log Buffer, or ingest into OpenSearch domain directly.
3. The Log Buffer triggers the Lambda function (log processor) to run.
4. The log processor reads and processes the log records and ingests the logs into the OpenSearch domain.
5. Logs that fail to be processed are exported to an Amazon S3 bucket (Backup Bucket).

AWS Well-Architected pillars

This solution was designed with best practices from the AWS Well-Architected Framework which helps customers design and operate reliable, secure, efficient, and cost-effective workloads in the cloud.

This section describes how the design principles and best practices of the Well-Architected Framework were applied when building this solution.

Operational excellence

This section describes how the principles and best practices of the operational excellence pillar were applied when designing this solution.

The solution pushes metrics, logs and traces to Amazon CloudWatch at various stages to provide observability into the infrastructure, Elastic load balancer, Amazon ECS cluster, Lambda functions, Step Function workflow and the rest of the solution components. This solution also creates the CloudWatch dashboards for each pipeline monitoring.

Security

This section describes how the principles and best practices of the security pillar were applied when designing this solution.

- The web console users are authenticated and authorized with Amazon Cognito or OpenID Connect.
- All inter-service communications use AWS IAM roles.
- All roles used by the solution follows least-privilege access. That is, it only contains minimum permissions required so the service can function properly.

Reliability

This section describes how the principles and best practices of the reliability pillar were applied when designing this solution.

- Using AWS serverless services wherever possible (for example, AWS AppSync, Amazon DynamoDB, AWS Lambda, AWS Step Functions, Amazon S3, and Amazon SQS) to ensure high availability and recovery from service failure.
- Configuration management content of the solution is stored in Amazon DynamoDB, all of your data is stored on solid-state disks (SSDs) and is automatically replicated across multiple Availability Zones (AZs) in an AWS Region, providing built-in high availability and data durability.

Performance efficiency

This section describes how the principles and best practices of the performance efficiency pillar were applied when designing this solution.

- The ability to launch this solution in any Region that supports AWS services in this solution such as: Amazon S3, Amazon ECS, Elastic load balancer.
- Using serverless architecture removes the need for you to run and maintain physical servers for traditional compute activities.
- Automatically testing and deploying this solution daily. Reviewing this solution by solution architects and subject matter experts for areas to experiment and improve.

Cost optimization

This section describes how the principles and best practices of the cost optimization pillar were applied when designing this solution.

- Use Autoscaling Group so that the compute costs are only related to how much data is ingested and processed.
- Using serverless services such as Amazon S3, Amazon DynamoDB, AWS Lambda, etc, so that customers only get charged for what they use.

Sustainability

This section describes how the principles and best practices of the sustainability pillar were applied when designing this solution.

- The solution's serverless design (using Amazon Kinesis Data Streams, Amazon S3, AWS Lambda) and the use of managed services (such as Amazon ECS) are aimed at reducing carbon footprint compared to the footprint of continually operating on-premises servers.

Architecture details

This section describes the components and AWS services that make up this solution and the architecture details on how these components work together.

Solution components

The solution consists of the following components:

Domain Management

This solution uses Amazon OpenSearch Service as the underlying engine to store and analyze logs. You can import an existing Amazon OpenSearch Service domain for log ingestion, and provide an access proxy to the Amazon OpenSearch Service dashboards within VPC. Moreover, you can set up recommended Amazon CloudWatch alarms for Amazon OpenSearch Service.

Analytics Pipelines

A log pipeline includes a series of log processing steps, including collecting logs from sources, processing and sending them to Amazon OpenSearch Service for further analysis. Centralized Logging with OpenSearch supports AWS Service log ingestion and server-side application log ingestion.

Service Log Pipeline

This solution supports out of the box log analysis for AWS service logs, such as Amazon S3 access logs, and Elastic Load Balancing access logs. The component is designed to reduce the complexities of building log analytics pipelines for different AWS services with different formats.

Application Log Pipeline

This solution supports out of the box log analysis for application logs, such as Nginx/Apache logs or general application logs via regex parser. The component uses Fluent Bit as the underlying log agent to collect logs from application servers, and allows you to easily install log agent and monitor the agent health via System Manager.

AWS services in this solution

The following AWS services are included in this solution:

AWS service	Description
Amazon CloudFront	To distribute the frontend web UI assets.
Amazon S3	To store the static web assets (frontend user interface), and also uses it as a data buffer for log shipping.
Amazon Cognito	To authenticate users (in AWS Regions).
AWS AppSync	To provide the backend GraphQL APIs.
Amazon DynamoDB	To store the solution related information as backend database.
AWS Lambda	To interact with other AWS Services to process core logic of managing log pipelines or log agents, and obtain information updated in DynamoDB tables.
AWS Step Functions	To orchestrate on-demand AWS CloudForm ation deployment of a set of predefined stacks for log pipeline management.
AWS CloudFormation	To provision the AWS resources for the modules of pipelines and the solution web console.
AWS Systems Manager	To manage log agents for collecting logs from application servers, such as installing log agents (Fluent Bit) for application servers.
Amazon Kinesis Data Streams	To subscribe to logs from a CloudWatch Log Group or as a data buffer for log shipping, and then initiate the log processor Lambda function to run.
Amazon Data Firehose	To subscribe the logs from CloudWatch Log Group and then put logs into Amazon S3.

AWS service	Description
Amazon SQS	To receive Amazon S3 Event Notifications and then initiate the log processor Lambda function to run.

Plan your deployment

This section describes the cost, security, Regions, and other considerations prior to deploying the solution.

Cost

> **Important**
>
> The following cost estimations are examples and may vary depending on your environment.

You will be responsible for the cost of the AWS services used when running the solution. The main factors affecting the solution cost include:

- Type of logs to be ingested
- Volume of logs to be ingested/processed
- Size of the log message
- Location of logs
- Additional features

As of this revision, the following examples demonstrate the cost estimation of 10/100/1000 GB daily log ingestion for running this solution with default settings in the US East (N. Virginia) Region. The total cost is composed of Log Analytics Engine Cost (**Amazon OpenSearch Service Cost** or **Light Engine Cost)**, **Solution Console Cost**, and **Additional Features Cost**.

Amazon OpenSearch Cost

Amazon OpenSearch Service Cost

- **OD**: On Demand
- **AURI_1**: All Upfront Reserved Instance 1 Year
- **Tiering**: The days stored in each tier. For example, 7H + 23W + 60C indicates that the log is stored in hot tier for 7 days, warm tier for 23 days, and cold tier for 60 days.
- **Replica**: The number of shard replicas.

Daily log Volum (GB)	Retent (days)	Tierin	Replic	OD Montl (USD)	AURI_ Montl (USD)	Dedica Maste	Data Node	EBS (GB)	UltraV Nodes	UltraV / Cold S3 Storag (GB)	OD cost per GB (USD)	AURI_1 cost per GB (USD)
10	30	30H	0	216.28	158.54	N/A	c6g.lar [2]	380	N/A	0	0.7209	0.52847
10	30	30H	1	289.35	223.94	N/A	m6g.la [2]	760	N/A	0	0.9645	0.74647
100	30	7H + 23W	0	989.49	825.97	m6g.la [3]	m6g.la [2]	886	mediu	0	0.3298	0.27532
100	30	7H + 23W	1	1295.8	1066.9	m6g.la [3]	m6g.la [4]	1772	mediu	0	0.4319	0.35564
100	90	7H + 23W + 60C	0	1133.4	969.97	m6g.la [3]	m6g.la [2]	886	mediu	8300	0.1259	0.10777
100	90	7H + 23W + 60C	1	1439.8	1210.9	m6g.la [3]	m6g.la [4]	1772	mediu	8300	0.1599	0.13455
100	180	7H + 23W + 150C	0	1349.4	1185.9	m6g.la [3]	m6g.la [2]	886	mediu	17300	0.0749	0.06589
100	180	7H + 23W + 150C	1	1655.8	1426.9	m6g.la [3]	m6g.la [4]	1772	mediu	17300	0.0919	0.07927

Daily log Volum (GB)	Reten (days)	Tierin	Replic	OD Montl (USD)	AURI_ Montl (USD)	Dedic Maste	Data Node	EBS (GB)	UltraV Nodes	UltraV / Cold S3 Storag (GB)	OD cost per GB (USD)	AURI_1 cost per GB (USD)
1000	30	7H + 23W	0	6101.1	5489.4	m6g.la [3]	r6g.xla e[6]	8856	mediu]	23000	0.2033	0.18298
1000	30	7H + 23W	1	8759.4	7635.8	m6g.la [3]	r6g.2x ge[6]	17712	mediu]	23000	0.2919	0.25453
1000	90	7H + 23W + 60C	0	8027.3	7245.4	m6g.la [3]	r6g.xla e[6]	8856	mediu]	83000	0.0891	0.0805
1000	90	7H + 23W + 60C	1	10199	9075.8	m6g.la [3]	r6g.2x ge[6]	17712	mediu]	83000	0.1133	0.10084
1000	180	7H + 23W + 150C	0	9701.1	9089.4	m6g.la [3]	r6g.xla e[6]	8856	mediu]	17300	0.0539	0.0505
1000	180	7H + 23W + 150C	1	12644	11420	m6g.la [3]	r6g.2x ge[6]	17712	mediu]	17300	0.0702	0.06345

Processing Cost

Log ingestion through Amazon S3

This section is applicable to:

- AWS service logs including Amazon S3 access logs, CloudFront standard logs, CloudTrail logs (S3), Elastic Load Balancing access logs, WAF logs, VPC Flow logs (S3), AWS Config logs, Amazon RDS/Aurora logs, and AWS Lambda Logs.
- Application Logs that use Amazon S3 as data buffer.

Assumptions:

- The logs stored in Amazon S3 are in gzip format.
- A 4MB compressed log file in S3 is roughly 100 MB in raw log size.
- A Lambda with 1 GB memory takes about 26 seconds to process a 4 MB compressed log file, namely 260 milliseconds (ms) per MB raw logs.
- The maximum compressed log file size is 5 MB.
- Ingesting logs from S3 will incur SQS and S3 request fees which are very low, or usually within the free tier.

You have N GB raw log per day, and the daily cost estimation is as follows:

When you use Lambda as log processor:

- Lambda Cost = 260 ms per MB x 1024 MB x N GB/day x $0.0000000167 per ms
- S3 Storage Cost = $0.023 per GB x NGB/day x 4% (compression)

When you use OSI as log processor:

- OSI Pipeline Cost = $0.24 per OCU per hour
- The maximum acount of S3 data 1 OCU can handle is around 20MB/s

The total monthly cost for ingesting AWS service logs is:

Total Monthly Cost (Lambda as processor) = (Lambda Cost + S3 Storage Cost) x 30 days

Daily Log Volume	Daily Lambda Cost (USD)	Daily S3 Storage Cost (USD)	Monthly Cost (USD)
10	$0.044	$0.009	$1.610
100	$0.445	$0.092	$16.099

Daily Log Volume	Daily Lambda Cost (USD)	Daily S3 Storage Cost (USD)	Monthly Cost (USD)
1000	$4.446	$0.920	$160.986
5000	$22.230	$4.600	$804.900

Total Monthly Cost (OSI as processor) = (OSI Cost + S3 Storage Cost) x 30 days

Daily Log Volume	Daily OSI Cost (USD)	Daily S3 Storage Cost (USD)	Monthly Cost (USD)
10	$5.760	$0.001	$173.1
100	$5.760	$0.009	$175.5
1000	$11.520	$0.920	$373.2
5000	$34.560	$4.600	$1174.8

For Amazon RDS/Aurora logs and AWS Lambda Logs that deliver to CloudWatch Logs, apart from the S3 and Lambda costs listed above, there is additional cost of using Firehose (KDF) to subscribe to the CloudWatch Logs Stream and put them into an Amazon S3 bucket, and KDF is charging for a 5KB increments (less than 5KB per record is billed as 5KB).

Assuming Log size is 0.2 KB per record, then the daily KDF cost is estimated as below:

- Firehose Cost = $0.029 per GB x N GB/day x (5KB/0.2 KB)

For example, for 1GB logs per day, the extra monthly cost of KDF is $21.75.

Important

If you want to save cost charged by Firehose, make sure you activate logs only when needed. For example, you can choose not to activate RDS general logs unless required.

Logs ingestion through Amazon Kinesis Data Streams

This section is applicable to:

- AWS Services Logs including CloudFront real-time logs, CloudTrail logs (CloudWatch), and VPC Flow logs (CloudWatch).
- Application Logs that use Amazon KDS as data buffer

> **Important**
>
> The cost estimation does not include the logging cost of service. For example, CloudFront real-time logs are charged based on the number of log lines generated ($0.01 for every 1,000,000 log lines). There are also logs delivery to CloudWatch charges for CloudTrail and VPC Flow logs that enabled CloudWatch Logging. Please check the service pricing for more details.

The cost estimation is based on the following assumptions and facts:

- The average log message size is 1 KB.
- The daily log volume is L GB.
- The Lambda processor memory is 1024 MB.
- Every Lambda invocation processes 1 MB logs.
- One Lambda invocation processes one shard of Kinesis, and Lambda can scale up to more concurrent invocations to process multiple shards.
- The Lambda runtime to process log less than 5 MB is 500ms.
- 30% additional shards are provided to handle traffic jitter.
- One Kinesis shard intake log size is = 1 MB /second x 3600 seconds per hour x 24 hours x 0.7 = 60.48 GB/day.
- The desired Kinesis Shard number S is = Round_up_to_next_integer(Daily log volume L / 60.48).

Based on the above assumptions, here is the daily cost estimation formula:

- Kinesis Shard Hour Cost = $0.015 / shard hour x 24 hours per day x S shards
- Kinesis PUT Payload Unit Cost = $0.014 per million units x 1 millions per GB x L GB per day
- Lambda Cost = $0.0000000167 per 1ms x 500 ms per invocation x 1,000 invocations per GB x L GB per day

Total Monthly Cost = (Kinesis Shard Hour Cost + Kinesis PUT Payload Unit Cost + Lambda Cost) x 30 days

Daily Log Volume (GB)	Shards	Daily Kinesis Shard Hour Cost (USD)	Daily Kinesis PUT Payload Unit Cost (USD)	Daily Lambda Cost (USD)	Monthly Cost (USD)
10	1	$0.36	$0.14	$0.0835	$17.505
100	2	$0.72	$1.40	$0.835	$88.65
1000	17	$6.12	$14.00	$8.35	$854.10

Light Engine Cost

Amazon Service	Monthly Cost (USD) Raw log: 10GB daily Query: 50GB daily	Monthly Cost (USD) Raw log: 100GB daily Query: 300GB daily	Monthly Cost (USD) Raw log: 1TB daily Query: 1TB daily
Amazon S3	$1.49	$19.98	$148.99
AWS Lambda	$0.37	$0.73	$1.10
Amazon SQS	$0.00	$0.00	$0.00
Amazon DynamoDB	$3.79	$3.79	$3.79
AWS Step Functions	$8.07	$16.14	$26.90
Amazon SNS	$0.18	$0.18	$0.18
Amazon Athena	$7.25	$43.51	$148.54
Amazon EC2	$29.20	$29.20	$29.20
Total	$50.35	$113.53	$358.70

Solution Console Cost

> **Note**
>
> AWS Step Functions, Amazon CloudWatch, AWS Systems Manager, and Amazon EventBridge are all within free-tier.

A web console is created automatically when you deploy the solution. Assume the visits to the console are 3,000 times in a month (30 days), it will incur the following cost:

Service	Monthly Cost (USD)
Amazon CloudFront (1GB Data Transfer Out to Internet and 1GB Data Transfer Out to Origin)	0.25
Amazon S3	0.027
Amazon Cognito	0.05
AWS AppSync	0.01
Amazon DynamoDB	1.00
AWS Lambda	0.132
Total	1.469

Additional Features Cost

> **Note**
>
> You will not be charged if you do not use the additional features in the Centralized Logging with OpenSearch console.

Access Proxy

If you deploy the Access Proxy through Centralized Logging with OpenSearch, additional charges will apply. The total cost varies depending on the instance type and number of instances. As of this revision, the following are two examples for the cost estimation in the US East (N. Virginia) Region.

Example 1: Instance Type - t3.nano, Instance Number - 2

- EC2 cost = t3.nano 1Y RI All Upfront price $26.28 x 2 / 12 months = $4.38/month
- EBS Cost = EBS $0.1 GB/month x 8 GB x 2 = $1.6/month (The EBS attached to the EC2 instance is 8 GB)
- Elastic Load Balancer Cost = $0.0225 per ALB-hour x 720 hours/month = $16.2/month

Total Monthly Cost = $4.38 EC2 Cost + $1.6 EBS Cost + $16.2 Elastic Load Balancer Cost = **$22.18**

Example 2: Instance Type - t3.large, Instance Number - 2

- EC2 Cost = t3.large 1Y RI All Upfront $426.612 x 2 / 12 months = $71.1/month
- EBS Cost = $0.1 GB/month x 8 GB x 2 = $1.6/month (The EBS attached to the EC2 instance is 8 GB)
- Elastic Load Balancer Cost = $0.0225 per ALB-hour x 720 hours/month = $16.2/month

Total Monthly Cost = $71.1 EC2 Cost + $1.6 EBS Cost + $16.2 Elastic Load Balancer Cost = **$88.9**

Amazon OpenSearch Service Alarms

If you deploy the alarms through Centralized Logging with OpenSearch, the Amazon CloudWatch Pricing will apply.

Pipeline Alarms

Log Type	Alarm Count	Number of Standard Resolution Alarm Metrics (USD)	Monthly Cost per Ingestion per Pipeline (USD)
AWS Service logs	4	$0.1	$0.4
Application logs	5	$0.1	$0.5

Pipeline Monitoring

Log processor

Assumptions:

- Deployment in the US East (N. Virginia) Region (us-east-1)
- A processor Lambda will be triggered every 60 seconds. The monthly metric put request number is 60 (requests) x 24 (hours) x 30 (days) = 43,200
- PutMetricData: 43,200 requests x 0.00001 USD = 0.432 USD
- There are 4 metrics for **Service Logs** (**total logs, failed logs, loaded logs, excluded logs**) and 3 metrics (**total logs, failed logs, loaded logs**) for **Application logs**
- Amazon CloudWatch Logs API = PutMetricData x Number of Metrics
- Amazon CloudWatch Logs Metric = Number of Metrics x 0.3

Log Type	Monthly Metric Put Request Number	Number of Metrics	Amazon CloudWatch Logs API (USD)	Amazon CloudWatch Logs Metric (USD)	Monthly Cost Per Source/Per Pipeline (USD)
AWS Service logs	43,200	4	$1.728	$1.20	$2.928
Application logs	43,200	3	$1.296	$0.90	$2.196

Fluent Bit

Assumptions:

- Deployment in the US East (N. Virginia) Region (us-east-1)
- There are 7 metrics: `FluentBitOutputProcRecords`, `FluentBitOutputProcBytes`, `FluentBitOutputDroppedRecords`, `FluentBitOutputErrors`, `FluentBitOutputRetriedRecords`, `FluentBitOutputRetriesFailed`, `FluentBitOutputRetries`. For more information, refer to the Monitoring section.

- Number of Metrics requested: an interval of 60 seconds to put logs from Fluent Bit to Amazon CloudWatch (60 requests in an hour). Monthly put requests are 60 (requests) x 24 (hours) x 30 (days) = 43,200
- PutMetricData: 43,200 requests x 0.00001 USD = 0.432 USD
- CloudWatch Logs API = PutMetricData x Number of Metrics x Number of Instances
- CloudWatch Logs Metric = Number of Metrics x 0.3

Number of EC2 Instances / EKS Nodes	Amazon CloudWatch Logs API (USD)	Amazon CloudWatch Logs Log Storage & Ingested (Calculated by AWS Pricing Calculator) (USD)	Amazon CloudWatch Logs Metric (USD)	Monthly Cost Per Source/Per Pipeline (USD)
1	$3.024	$0.04	$2.10	$5.164
10	$30.24	$0.35	$2.10	$32.69
100	$302.40	$3.53	$2.10	$308.03

How to view main stack and pipeline cost

Activating user-defined cost allocation tags

For tags to appear on your billing reports, you must activate them. The user-defined cost allocation tags represent the tag key, which you activate in the Billing and Cost Management console. Once you activate or deactivate the tag key, it will affect all tag values that share the same tag key. A tag key can have multiple tag values. For more information, see AWS Billing and Cost Management API Reference.

How to activate your tag keys

1. Sign in to the AWS Management Console and open the AWS Billing and Cost Management console.

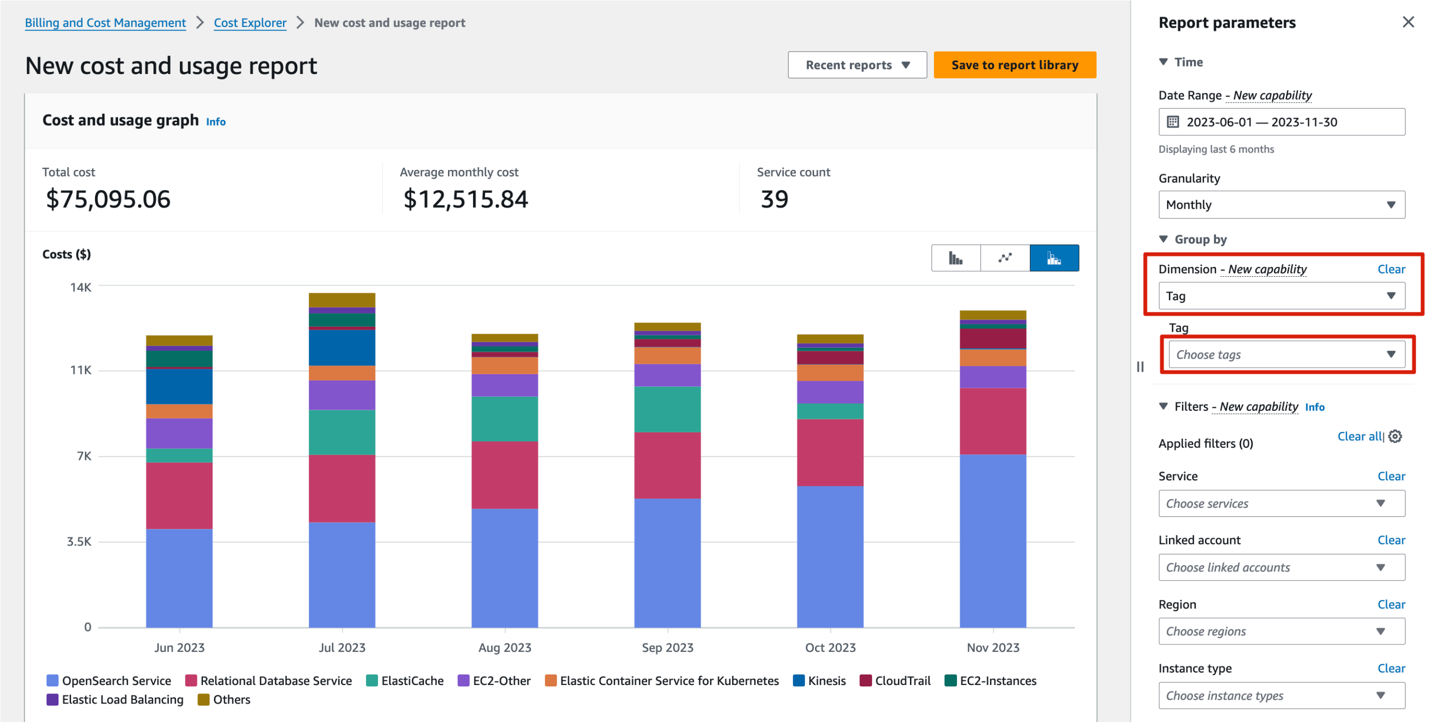
2. In the navigation pane, choose **Cost Allocation Tags** under **Cost Organization**.
3. Select the tage keys **CLOSolutionCostAnalysis** to activate.
4. Choose **Activate**.

> **Note**
>
> After you create and apply user-defined tags to your resources, it can take up to 24 hours for the tag keys to appear on your cost allocation tags page for activation.

For an example of how tag keys appear in your billing report with cost allocation tags, see Viewing a cost allocation report.

How to view cost explorer dashboard

1. Sign in to the AWS Management Console and open the AWS Billing and Cost Management console.
2. In the navigation pane, choose **Cost Explorer**.
3. Choose **Tag** as the displayed Dimension and select the specific tag **CLOSolutionCostAnalysis** to filter.
4. Try later if your activated tag is absent in the dropdown list. This may indicate that the activation process is still in progress, and it can take up to 24 hours for tag keys to activate.



Security

When you build systems on AWS infrastructure, security responsibilities are shared between you and AWS. This shared responsibility model reduces your operational burden because AWS operates, manages, and controls the components including the host operating system, the virtualization layer, and the physical security of the facilities in which the services operate. For more information about AWS security, see AWS Cloud Security.

IAM Roles

AWS Identity and Access Management (IAM) roles allow customers to assign granular access policies and permissions to services and users on the AWS Cloud. This solution creates IAM roles that grant the solution's AWS Lambda functions, AWS AppSync and Amazon Cognito access to create regional resources.

Security Groups

The security groups created in this solution are designed to control and isolate network traffic between the solution components. We recommend that you review the security groups and further restrict access as needed once the deployment is up and running.

Amazon CloudFront

This solution deploys a web console hosted in an Amazon Simple Storage Service (Amazon S3) bucket. To help reduce latency and improve security, this solution includes an Amazon CloudFront distribution with an origin access identity, which is a CloudFront user that provides public access to the solution's website bucket contents. For more information, refer to Restricting Access to Amazon S3 Content by Using an Origin Access Identity in the Amazon CloudFront Developer Guide.

Amazon EC2

This solution creates a Nginx based proxy, which will allow you to access the OpenSearch provisioned within VPC environment. The Nginx is hosted using EC2 instances. We recommend you to use AWS Systems Manager Patch Manager to patch the instances periodically. Patch Manager is a capability of AWS Systems Manager that automates the process of patching managed nodes with updates. You can choose to show only a report of missing patches (a Scan operation), or to automatically install all patches which are missing (a Scan and install operation).

Supported AWS Regions

This solution uses services which may not be currently available in all AWS Regions. Launch this solution in an AWS Region where required services are available. For the most current availability by Region, refer to the AWS Regional Services List.

Centralized Logging with OpenSearch provides two types of authentication, Cognito User Pool and OpenID Connect (OIDC) Provider. You must choose to launch the solution with OpenID Connect if one of the following cases occurs:

- Cognito User Pool is not available in your AWS Region.
- You already have an OpenID Connect Provider and want to authenticate against it.

Supported regions for deployment

Region Name	Launch with Cognito User Pool	Launch with OpenID Connect
US East (N. Virginia)	✓	✓
US East (Ohio)	✓	✓
US West (N. California)	✓	✓

Region Name	Launch with Cognito User Pool	Launch with OpenID Connect
US West (Oregon)	✓	✓
Africa (Cape Town)	✓	✓
Asia Pacific (Hong Kong)	✓	✓
Asia Pacific (Mumbai)	✓	✓
Asia Pacific (Osaka)	✓	✓
Asia Pacific (Seoul)	✓	✓
Asia Pacific (Singapore)	✓	✓
Asia Pacific (Sydney)	✓	✓
Asia Pacific (Tokyo)	✓	✓
Asia Pacific (Hyderabad)	✓	✓
Asia Pacific (Jakarta)	✓	✓
Asia Pacific (Melbourne)	✓	✓
Israel (Tel Aviv)	✓	✓
Canada (Central)	✓	✓
Canada (Calgary)	✓	✓
Europe (Frankfurt)	✓	✓
Europe (Ireland)	✓	✓
Europe (London)	✓	✓
Europe (Milan)	✓	✓
Europe (Paris)	✓	✓

Region Name	Launch with Cognito User Pool	Launch with OpenID Connect
Europe (Stockholm)	✓	✓
Europe (Spain)	✓	✓
Europe (Zurich)	✓	✓
Middle East (Bahrain)	✓	✓
Middle East (UAE)	X	✓
South America (Sao Paulo)	✓	✓
China (Beijing) Region Operated by Sinnet	X	✓
China (Ningxia) Regions operated by NWCD	X	✓

Important

You can have only one active Centralized Logging with OpenSearch solution stack in one region. If your deployment failed, make sure you have deleted the failed stack before retrying the deployment.

Automated deployment

Before you launch the solution, review the architecture, supported regions, and other considerations discussed in this guide. Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Prerequisites

Review all the considerations and make sure you have the following in the target region you want to deploy the solution:

- At least one vacancy to create new VPCs, if you choose to launch with new VPC.
- At least two vacant Elastic IP (EIP) addresses, if you choose to launch with new VPC.
- At least four vacant S3 buckets.

Deployment in AWS Regions

Centralized Logging with OpenSearch provides two ways to authenticate and log into the Centralized Logging with OpenSearch console. For some AWS regions where Cognito User Pool is not available (for example, Hong Kong), you need to launch the solution with OpenID Connect provider.

- Launch with Cognito User Pool
- Launch with OpenID Connect

For more information about supported regions, see Regional deployments.

Deployment in AWS China Regions

AWS China Regions do not have Cognito User Pool. You must launch the solution with OpenID Connect.

- Launch with OpenID Connect

Launch with Cognito User Pool

Time to deploy: Approximately 15 minutes

Deployment Overview

Use the following steps to deploy this solution on AWS.

[Step 1. Launch the stack](#)

[Step 2. Launch the web console](#)

Step 1. Launch the stack

This AWS CloudFormation template automatically deploys the Centralized Logging with OpenSearch solution on AWS.

1. Sign in to the AWS Management Console and select the button to launch the AWS CloudFormation template.

	Launch in AWS Management Console
Launch with a new VPC	Launch Stack
Launch with an existing VPC	Launch Stack

2. The template is launched in the default region after you log in to the console. To launch the Centralized Logging with OpenSearch solution in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL is shown in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, refer to IAM and STS Limits in the *AWS Identity and Access Management User Guide*.
5. Under **Parameters**, review the parameters for the template and modify them as necessary.
   - If you are launching the solution in a new VPC, this solution uses the following parameters:

Parameter	Default	Description
Admin User Email	`Requires input`	Specify the email of the Administrator. This email

Parameter	Default	Description
		address will receive a temporary password to access the Centralized Logging with OpenSearch web console. You can create more users directly in the provisioned Cognito User Pool after launching the solution.

- If you are launching the solution in an existing VPC, this solution uses the following parameters:

Parameter	Default	Description
Admin User Email	*`Requires input`*	Specify the email of the Administrator. This email address will receive a temporary password to access the Centralized Logging with OpenSearch web console. You can create more users directly in the provisioned Cognito User Pool after launching the solution.
VPC ID	*`Requires input`*	Specify the existing VPC ID in which you are launching the Centralized Logging with OpenSearch solution.

Parameter	Default	Description
Public Subnet IDs	*Requires input*	Specify the two public subnets in the selected VPC. The subnets must have routes point to an Internet Gateway.
Private Subnet IDs	*Requires input*	Specify the two private subnets in the selected VPC. The subnets must have routes point to an NAT Gateway.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Add new tag** and enter the following key and value:
   - Key: CLOSolutionCostAnalysis
   - Value: CLOSolutionCostAnalysis

   You can activate the CLOSolutionCostAnalysis tag after all resources has been successfully deployed.
8. Choose **Next**.
9. On the **Review** page, review and confirm the settings. Select the box acknowledging that the template creates AWS Identity and Access Management (IAM) resources.
10. Choose **Create stack** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately 15 minutes.

Step 2. Launch the web Console

After the stack is successfully created, this solution generates a CloudFront domain name that gives you access to the Centralized Logging with OpenSearch web console. Meanwhile, an auto-generated temporary password (excluding the last digit .) will be sent to your email address.

1. Sign in to the AWS CloudFormation console.

2. On the **Stacks** page, select the solution's stack.
3. Choose the **Outputs** tab and record the domain name.
4. Open the **WebConsoleUrl** using a web browser, and navigate to a sign-in page.
5. Enter the **Email** and the temporary password.

   a. Set a new account password.

   b. (Optional) Verify your email address for account recovery.
6. After the verification is complete, the system opens the Centralized Logging with OpenSearch web console.

Once you have logged into the Centralized Logging with OpenSearch console, you can import an Amazon OpenSearch Service domain and build log analytics pipelines.

Launch with OpenID Connect (OIDC)

Time to deploy: Approximately 30 minutes

Prerequisites

> **Important**
>
> The Centralized Logging with OpenSearch console is served via CloudFront distribution which is considered as an Internet information service. If you are deploying the solution in **AWS China Regions**, the domain must have a valid ICP Recordal.

- A domain. You will use this domain to access the Centralized Logging with OpenSearch console (Required for AWS China Regions, optional for AWS Regions).
- An SSL certificate in AWS IAM. The SSL must be associated with the given domain. Follow this guide to upload SSL certificate to IAM. Note that this is required for AWS China Regions, but is not recommended for AWS Regions.
- Make sure to request or import the ACM certificate in the US East (N. Virginia) Region (us-east-1). Note that this is not required for AWS China Regions, and is optional for AWS Regions.

Deployment Overview

Use the following steps to deploy this solution on AWS.

Step 1. Create OIDC client

Step 2. Launch the stack

Step 3. Setup DNS Resolver

Step 4. Launch the web console

Step 1. Create OIDC client

You can use different kinds of OpenID Connector (OIDC) providers. This section introduces Option 1 to Option 4.

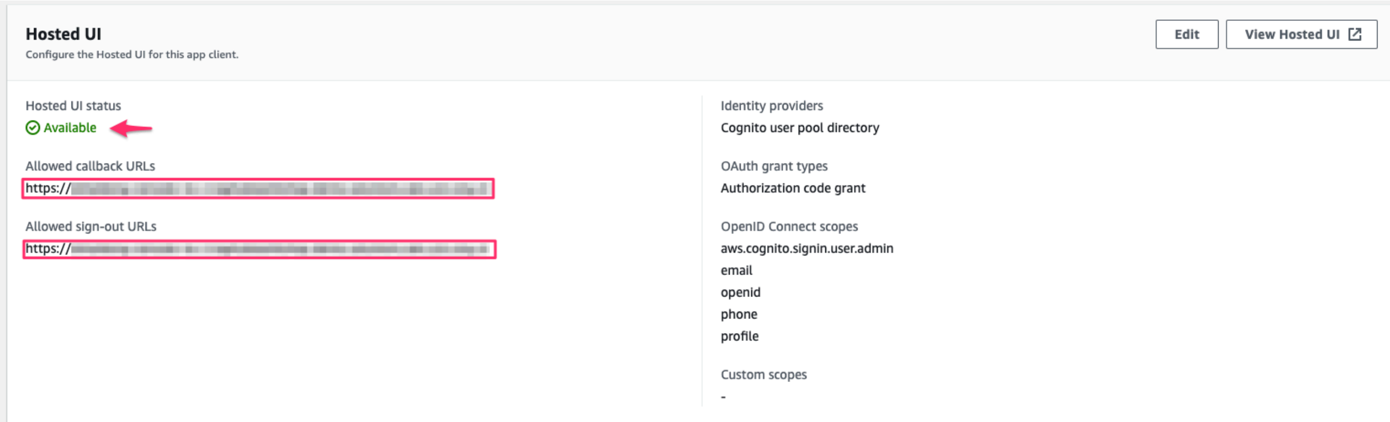
- (Option 1) Using Amazon Cognito from another region as OIDC provider.
- (Option 2) Authing, which is an example of a third-party authentication provider.
- (Option 3) Keycloak, which is a solution maintained by AWS and can serve as an authentication identity provider.
- (Option 4) ADFS, which is a service offered by Microsoft.
- (Option 5) Other third-party authentication platforms such as Auth0.

Follow the steps below to create an OIDC client, and obtain the `client_id` and `issuer`.

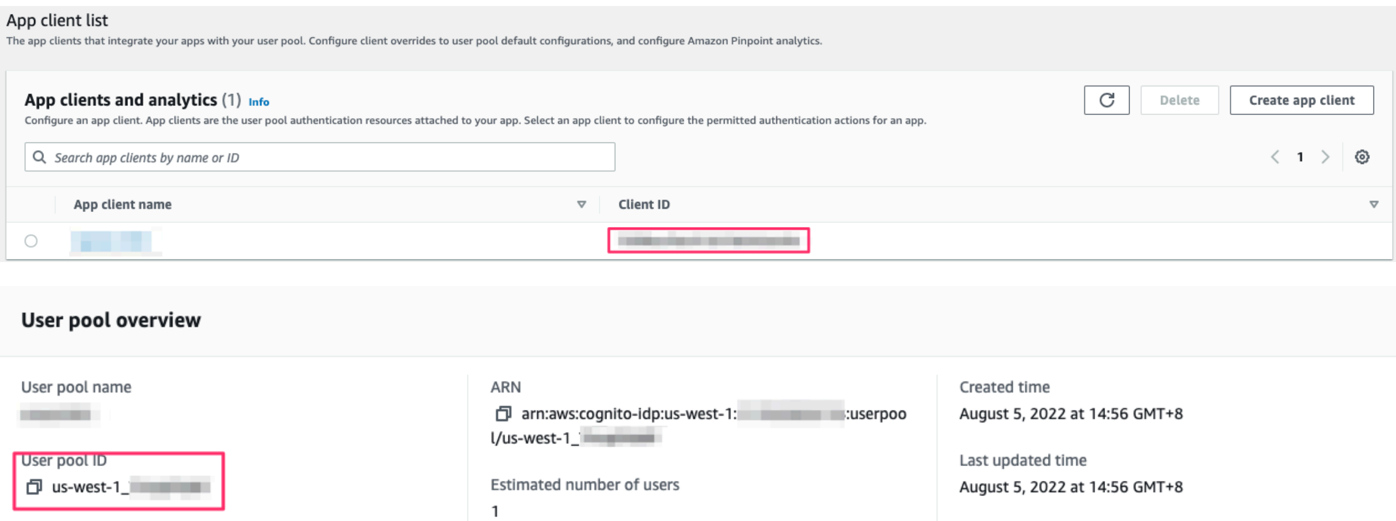
(Option 1) Using Cognito User Pool from another region

You can leverage the Cognito User Pool in a supported AWS Standard Region as the OIDC provider.

1. Go to the Amazon Cognito console in an AWS Region.
2. Set up the hosted UI with the Amazon Cognito console based on this guide.
3. Choose **Public client** when selecting the **App type**.
4. Enter the **Callback URL** and **Sign out URL** using your domain name for Centralized Logging with OpenSearch console. If your hosted UI is set up, you should be able to see something like below.



5. Save the App client ID, User pool ID and the AWS Region to a file, which will be used later.

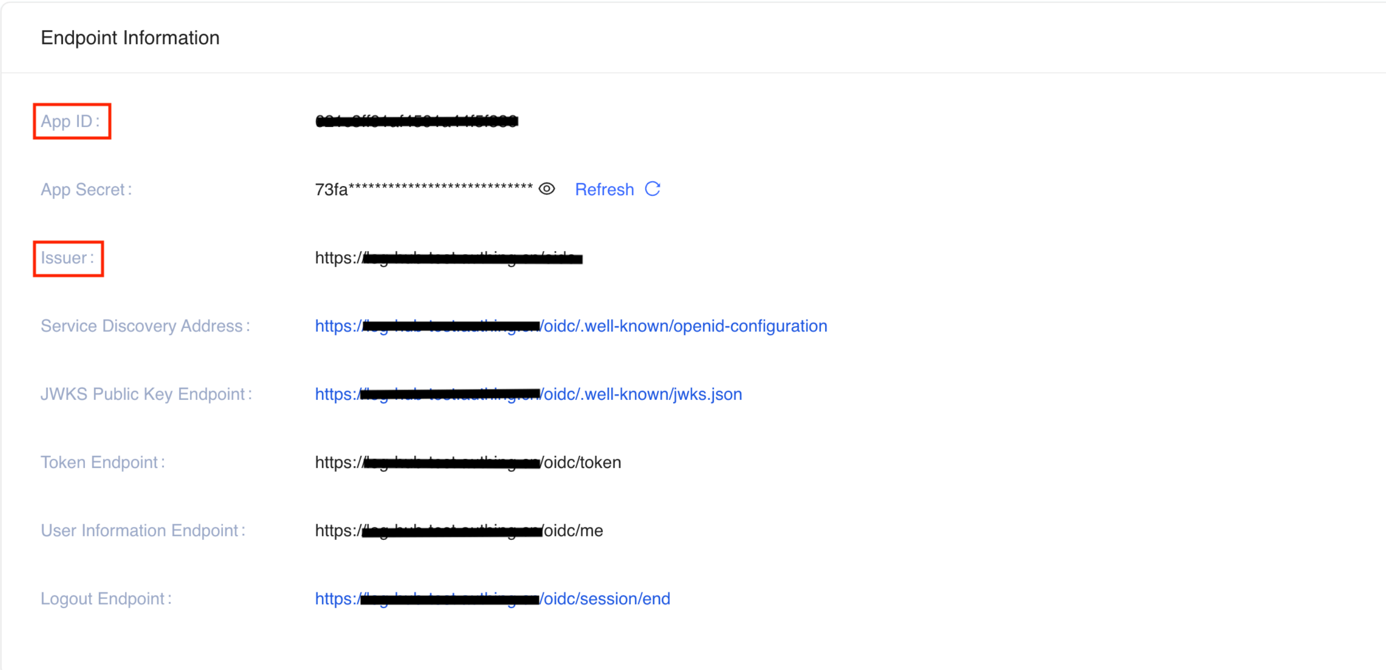


In Step 2. Launch the stack, the OidcClientID is the `App client ID`, and OidcProvider is `https://cognito-idp.${REGION}.amazonaws.com/${USER_POOL_ID}`.

(Option 2) Authing.cn OIDC client

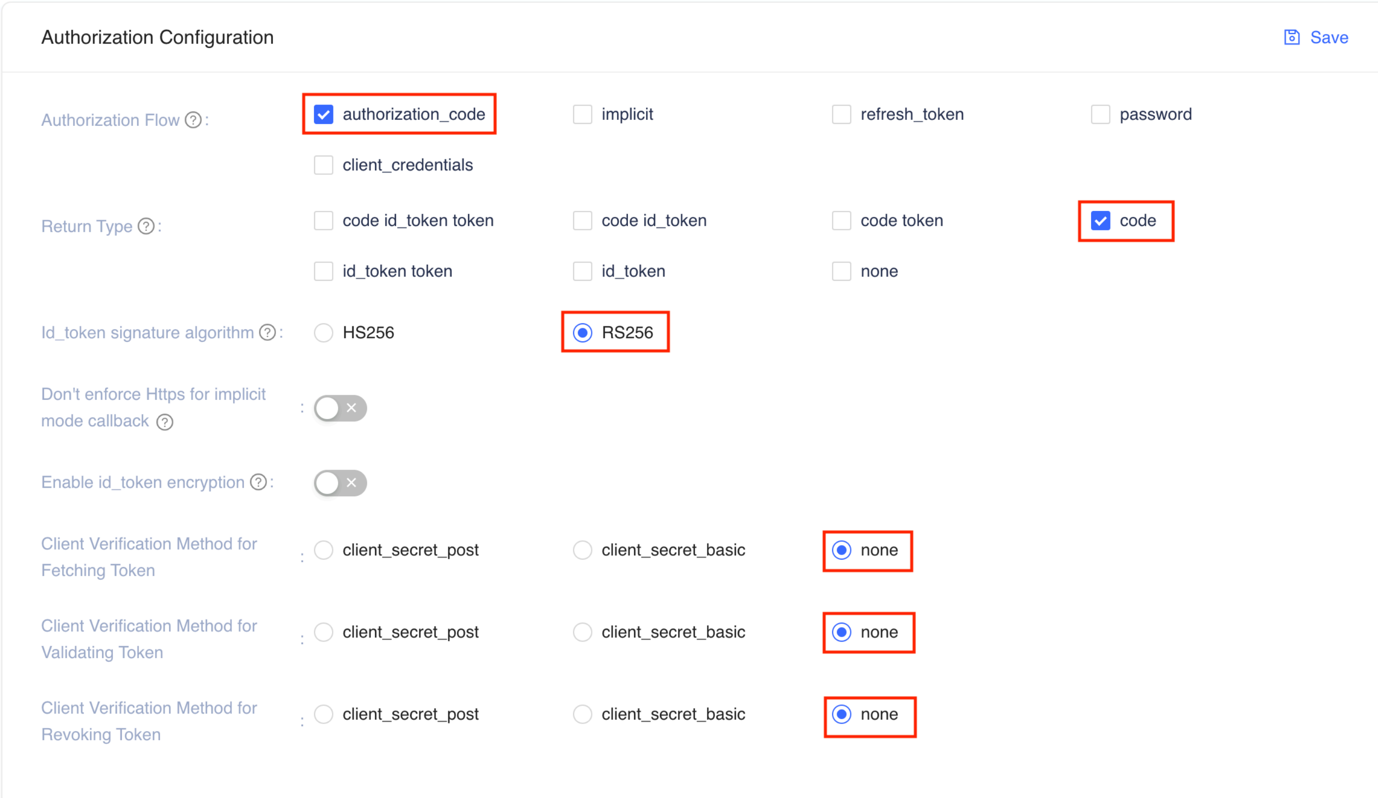
1. Go to the Authing console.
2. Create a user pool if you don't have one.
3. Select the user pool.
4. On the left navigation bar, select **Self-built App** under **Applications**.
5. Click the **Create** button.
6. Enter the **Application Name**, and **Subdomain**.

7. Save the `App ID` (that is, `client_id`) and `Issuer` to a text file from Endpoint Information, which will be used later.



8. Update the `Login Callback URL` and `Logout Callback URL` to your IPC recorded domain name.

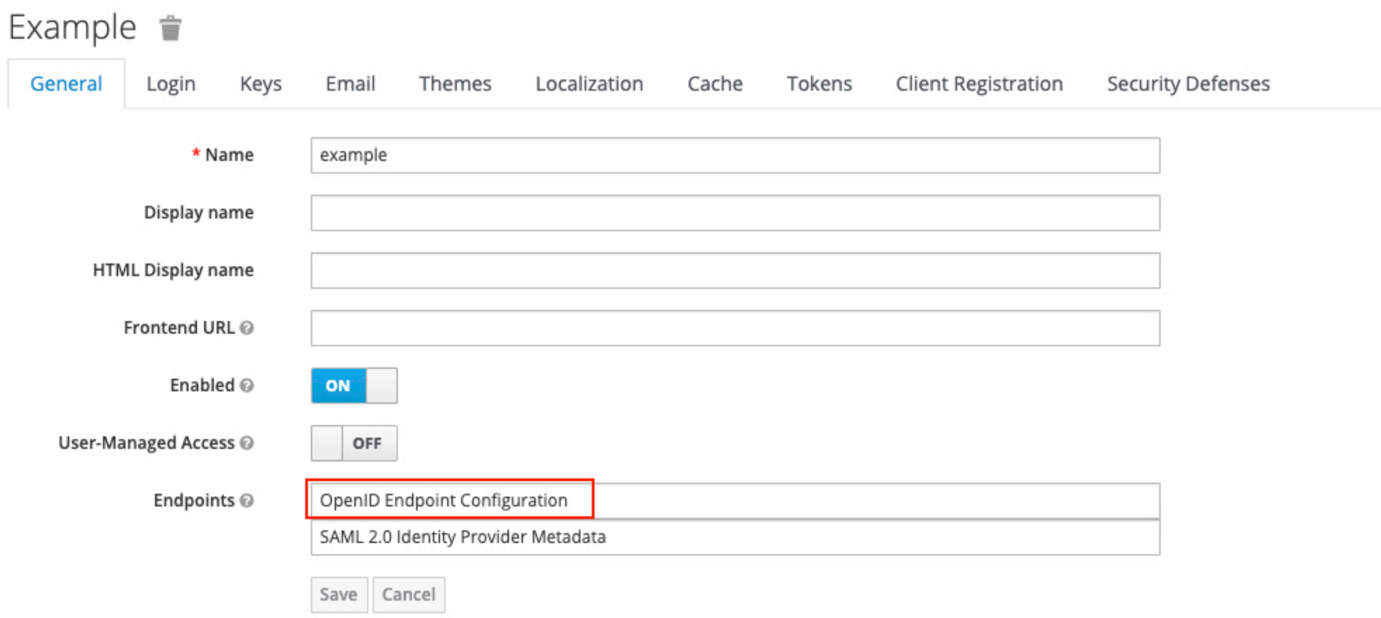
9. Set the Authorization Configuration.



You have successfully created an authing self-built application.

(Option 3) Keycloak OIDC client

1. Deploy the Keycloak solution in AWS China Regions following [this guide].
2. Sign in to the Keycloak console.
3. On the left navigation bar, select **Add realm**. Skip this step if you already have a realm.
4. Go to the realm setting page. Choose **Endpoints**, and then **OpenID Endpoint Configuration** from the list.



5. In the JSON file that opens up in your browser, record the **issuer** value which will be used later.

6. Go back to Keycloak console and select **Clients** on the left navigation bar, and choose **Create**.
7. Enter a Client ID, which must contain 24 letters (case-insensitive) or numbers. Record the **Client ID** which will be used later.
8. Change client settings. Enter `https://<Centralized Logging with OpenSearch Console domain>` in **Valid Redirect URIs**, and enter * and + in **Web Origins**.
9. In the Advanced Settings, set the **Access Token Lifespan** to at least 5 minutes.
10. Select **Users** on the left navigation bar.
11. Click **Add user** and enter **Username**.
12. After the user is created, select **Credentials**, and enter **Password**.

The issuer value is `https://<KEYCLOAK_DOMAIN_NAME>/auth/realms/<REALM_NAME>`.

(Option 4) ADFS OpenID Connect Client

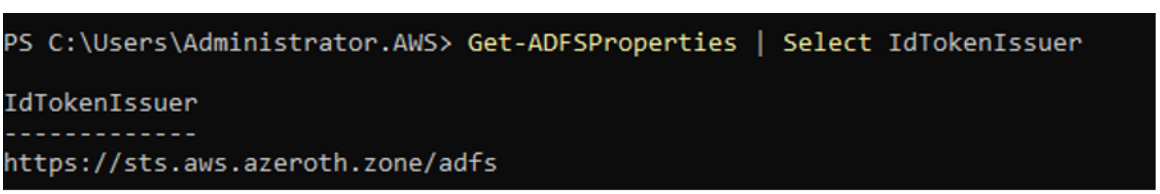
1. Make sure your ADFS is installed. For information about how to install ADFS, refer to this guide.

2. Make sure you can log in to the ADFS Sign On page. The URL should be `https://adfs.domain.com/adfs/ls/idpinitiatedSignOn.aspx`, and you need to replace **adfs.domain.com** with your real ADFS domain.
3. Log on your **Domain Controller**, and open **Active Directory Users and Computers**.
4. Create a **Security Group** for Centralized Logging with OpenSearch Users, and add your planned Centralized Logging with OpenSearch users to this Security Group.
5. Log on to ADFS server, and open **ADFS Management**.
6. Right click **Application Groups**, choose **Application Group**, and enter the name for the Application Group. Select **Web browser accessing a web application** option under **Client-Server Applications**, and choose **Next**.
7. Record the **Client Identifier** (`client_id`) under **Redirect URI**, enter your Centralized Logging with OpenSearch domain (for example, `xx.domain.com`), and choose **Add**, and then choose **Next**.
8. In the **Choose Access Control Policy** window, select **Permit specific group**, choose **parameters** under Policy part, add the created Security Group in Step 4, then click **Next**. You can configure other access control policy based on your requirements.
9. Under Summary window, choose **Next**, and choose **Close**.
10. Open the Windows PowerShell on ADFS Server, and run the following commands to configure ADFS to allow CORS for your planned URL.

```
Set-AdfsResponseHeaders -EnableCORS $true
Set-AdfsResponseHeaders -CORSTrustedOrigins https://<your-centralized-logging-with-opensearch-domain>
```

11. Under Windows PowerShell on ADFS server, run the following command to get the Issuer (`issuer`) of ADFS, which is similar to `https://adfs.domain.com/adfs`.

```
Get-ADFSProperties | Select IdTokenIssuer
```



Step 2. Launch the stack

> **Important**
>
> You can only have one active Centralized Logging with OpenSearch solution stack in one region of an AWS account. If your deployment failed (for example, not meeting the requirements in prerequisites), make sure you have deleted the failed stack before retrying the deployment.

1. Sign in to the AWS Management Console and use the button below to launch the AWS CloudFormation template.

	Launch in AWS Management Console
Launch with a new VPC in AWS Regions	Launch Stack
Launch with an existing VPC in AWS Regions	Launch Stack
Launch with a new VPC in AWS China Regions	Launch Stack
Launch with an existing VPC in AWS China Regions	Launch Stack

2. The template is launched in the default region after you log in to the console. To launch the Centralized Logging with OpenSearch solution in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack. For information about naming character limitations, refer to IAM and STS Limits in the *AWS Identity and Access Management User Guide*.
5. Under **Parameters**, review the parameters for the template and modify them as necessary.
    - If you are launching the solution in a new VPC, this solution uses the following parameters:

Parameter	Default	Description
OidcClientId	*Requires input*	OpenID Connector client Id.
OidcProvider	*Requires input*	OpenID Connector provider issuer. The issuer must begin with `https://`
Domain	<Optional>	Custom domain for Centralized Logging with OpenSearch console. Do NOT add `http(s)` prefix.
IamCertificateID	<Optional>	The ID of the SSL certificate in IAM. The ID is composed of 21 characters of capital letters and digits. Use the list-server-certificates command to retrieve the ID.
AcmCertificateArn	<Optional>	Arn for ACM certificates requested (or imported) the certificate in the US East (N. Virginia) Region (us-east-1).

- If you are launching the solution in an existing VPC, this solution uses the following parameters:

Parameter	Default	Description
OidcClientId	*Requires input*	OpenID Connector client Id.
OidcProvider	*Requires input*	OpenID Connector provider issuer. The issuer must begin with https://

Parameter	Default	Description
Domain	<Optional>	Custom domain for Centralized Logging with OpenSearch console. Do NOT add http(s) prefix.
IamCertificateID	<Optional>	The ID of the SSL certificate in IAM. The ID is composed of 21 characters of capital letters and digits. Use the `list-server-certificates` command to retrieve the ID.
AcmCertificateArn	<Optional>	Arn for ACM certificates requested (or imported) the certificate in the US East (N. Virginia) Region (us-east-1).
VPC ID	*Requires input*	Specify the existing VPC ID in which you are launching the solution.
Public Subnet IDs	*Requires input*	Specify the two public subnets in the selected VPC. The subnets must have routes pointing to an Internet Gateway.
Private Subnet IDs	*Requires input*	Specify the two private subnets in the selected VPC. The subnets must have routes pointing to an NAT Gateway.

> **Important**
>
> - If you are deploying the solution in **AWS China Regions**, you must enter Domain, and IamCertificateID.
> - If you are deploying the solution in **AWS Regions**,
>   - when a custom domain name is required, you must enter Domain, and AcmCertificateArn.
>   - when no custom domain name is required, leave it blank for Domain, IamCertificateID, and AcmCertificateArn.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Add new tag** and enter the following key and value:
   - Key: CLOSolutionCostAnalysis
   - Value: CLOSolutionCostAnalysis

   You can activate the CLOSolutionCostAnalysis tag after all resources has been successfully deployed.
8. Choose **Next**.
9. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template creates AWS Identity and Access Management (IAM) resources.
10. Choose **Create stack** to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately 15 minutes.

Step 3. Setup DNS Resolver

This solution provisions a CloudFront distribution that gives you access to the Centralized Logging with OpenSearch console.

1. Sign in to the [AWS CloudFormation console](https://console.aws.amazon.com/cloudformation/).
2. Select the solution's stack.
3. Choose the **Outputs** tab.

4. Obtain the **WebConsoleUrl** as the endpoint.
5. Create a CNAME record in DNS resolver, which points to the endpoint address.

Step 4. Launch the web console

> **Important**
>
> You login credentials is managed by the OIDC provider. Before signing in to the Centralized Logging with OpenSearch console, make sure you have created at least one user in the OIDC provider's user pool.

1. Use the previous assigned CNAME to open the **OIDC Customer Domain URL** using a web browser.
2. Choose **Sign in to Centralized Logging with OpenSearch**, and navigate to OIDC provider.
3. Enter sign-in credentials. You may be asked to change your default password for first-time login, which depends on your OIDC provider's policy.
4. After the verification is complete, the system opens the Centralized Logging with OpenSearch web console.

Once you have logged into the Centralized Logging with OpenSearch console, you can import an Amazon OpenSearch Service domain and build log analytics pipelines.

Getting Started

After deploying the solution, refer to this section to quickly learn how to leverage Centralized Logging with OpenSearch for log ingestion (AWS CloudTrail logs as an example), and log visualization.

You can also choose to start with Domain management , then build AWS Service Log Analytics Pipelines and Application Log Analytics Pipelines.

Steps

- Step 1: Import an Amazon OpenSearch Service domain. Import an existing Amazon OpenSearch Service domain into the solution.
- Step 2: Create Access Proxy. Create a public access proxy which allows you to access the templated dashboard from anywhere.
- Step 3: Ingest CloudTrail Logs. Ingest CloudTrail logs into the specified Amazon OpenSearch Service domain.
- Step 4: Access built-in dashboard. View the dashboard of CloudTrail logs.

Step 1: Import an Amazon OpenSearch Service domain

To use the Centralized Logging with OpenSearch solution for the first time, you must import Amazon OpenSearch Service domains first.

Centralized Logging with OpenSearch supports Amazon OpenSearch Service domain with fine-grained access control enabled within a VPC only.

Important

Currently, Centralized Logging with OpenSearch supports Amazon OpenSearch Service with OpenSearch 1.3 or later.

Prerequisite

At least one Amazon OpenSearch Service domain within VPC. If you don't have an Amazon OpenSearch Service domain yet, you can create an Amazon OpenSearch Service domain within VPC. See Launching your Amazon OpenSearch Service domains within a VPC.

Steps

Use the following procedure to import an Amazon OpenSearch Service domain through the Centralized Logging with OpenSearch console.

1. Sign in to the Centralized Logging with OpenSearch console.
2. In the navigation pane, under **Domains**, choose **Import OpenSearch Domain**.
3. On the **Step 1. Select domain** page, choose a domain from the dropdown list.
4. Choose **Next**.
5. On the **Step 2. Configure network** page, under **Network creation**, choose **Automatic**. If your Centralized Logging with OpenSearch and OpenSearch domains reside in two different VPCs, the *Automatic* mode will create a VPC Peering Connection between them, and update route tables. See details in Set up VPC Peering.
6. On the **Step 3. Create tags** page, choose **Import**.

Step 2: Create Access Proxy

Note

Access proxy is optional and it incurs additional cost. If you can connect to Amazon OpenSearch Service's VPC (such as through VPN connection), you don't need to activate access proxy. You need to use it only if you want to connect to Amazon OpenSearch Service dashboard from public Internet.

You can create a Nginx proxy and create an DNS record pointing to the proxy, so that you can access the Amazon OpenSearch Service dashboard securely from public network. For more information, refer to Access Proxy in the Domain Management chapter.

Create a Nginx proxy

1. Sign in to the Centralized Logging with OpenSearch console.

2. In the navigation pane, under **Domains**, choose **OpenSearch domains**.
3. Select the domain from the table.
4. Under **General configuration**, choose **Enable** at the **Access Proxy** label.
5. On the **Create access proxy** page, under **Public access proxy**, select at least 2 subnets which contain `LogHubVpc/DefaultVPC/publicSubnetX` for the **Public Subnets**.
6. For **Public Security Group**, choose the Security Group which contains `ProxySecurityGroup`.
7. Enter the **Domain Name**.
8. Choose the associated **Load Balancer SSL Certificate** which applies to the domain name.
9. Choose the **Nginx Instance Key Name**.
10. Choose **Create**.

After provisioning the proxy infrastructure, you need to create an associated DNS record in your DNS resolver. The following introduces how to find the Application Load Balancer (ALB) domain, and then create a CNAME record pointing to this domain.

Create an DNS record

1. Sign in to the Centralized Logging with OpenSearch console.
2. In the navigation pane, under **Domains**, choose **OpenSearch domains**.
3. Select the domain from the table.
4. Choose the **Access Proxy** tab. Find **Load Balancer Domain**, which is the ALB domain.
5. Go to the DNS resolver, and create a CNAME record pointing to this domain. If your domain is managed by Amazon Route 53, refer to Creating records by using the Amazon Route 53 console.

Step 3: Ingest AWS CloudTrail Logs

You can build a log analytics pipeline to ingest AWS CloudTrail logs.

> **Important**
>
> Make sure your CloudTrail and Centralized Logging with OpenSearch are in the same AWS Region.

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the navigation pane, select **AWS Service Log Analytics Pipelines**.

3. Choose **Create a log ingestion**.
4. In the **AWS Services** section, choose **AWS CloudTrail**.
5. Choose **Next**.
6. Under **Specify settings**, for **Trail**, select one from the dropdown list.
7. Choose **Next**.
8. In the **Specify OpenSearch domain** section, select the imported domain for **Amazon OpenSearch Service domain**.
9. Choose **Yes** for **Sample dashboard**.
10. Keep default values and choose **Next**.
11. Choose **Create**.

Step 4: Access built-in Dashboard

After the DNS record takes effect, you can access the built-in dashboard from anywhere via proxy.

1. Enter the domain of the proxy in your browser. Alternatively, click the **Link** button under **Access Proxy** in the **General Configuration** section of the domain.
2. Enter your credentials to log in to Amazon OpenSearch Service Dashboard.
3. Click the username icon of Amazon OpenSearch Service dashboard from the top right corner.
4. Choose **Switch Tenants**.
5. On the **Select your tenant** page, choose **Global**, and click **Confirm**.
6. On the left navigation panel, choose **Dashboards**.
7. Choose the dashboard created automatically and start to explore your data.

Domain Management

This chapter describes how to manage Amazon OpenSearch Service domains through the Centralized Logging with OpenSearch console. An Amazon OpenSearch Service domain is synonymous with an Amazon OpenSearch Service cluster.

In this chapter, you will learn:

- Import & remove an Amazon OpenSearch Service Domain
- Create an access proxy
- Create recommended alarms

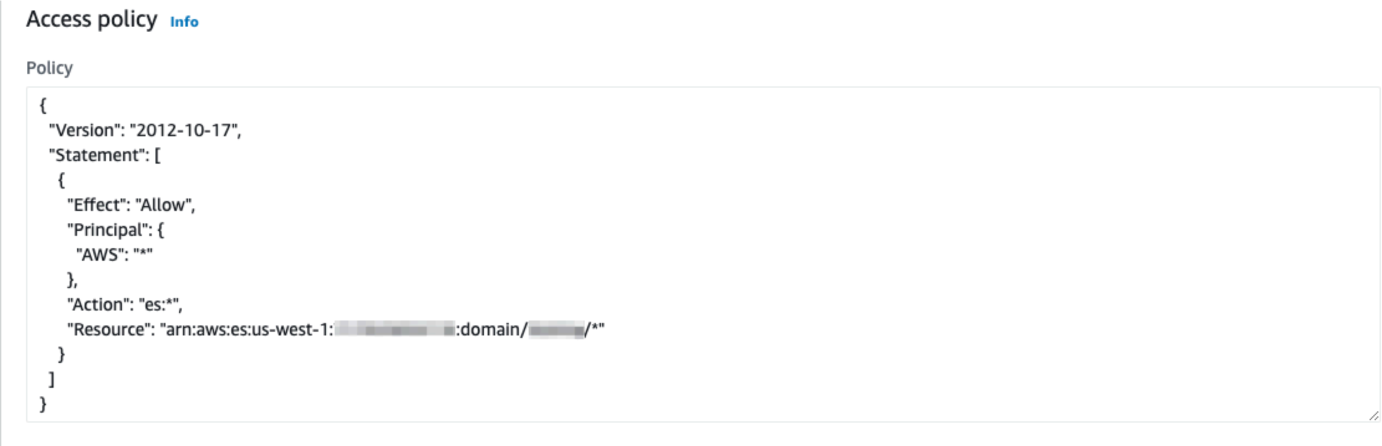
You can read the Getting Started chapter first and walk through the basic steps for using the Centralized Logging with OpenSearch solution.

Domain Operations

Once logged into the Centralized Logging with OpenSearch console, you can import an Amazon OpenSearch Service domain.

Prerequisite

1. Centralized Logging with OpenSearch supports Amazon OpenSearch Service, engine version OpenSearch 1.3 or later.
2. Centralized Logging with OpenSearch supports OpenSearch clusters within VPC. If you don't have an Amazon OpenSearch Service domain yet, you can create an Amazon OpenSearch Service domain within VPC. See Launching your Amazon OpenSearch Service domains within a VPC.
3. Centralized Logging with OpenSearch supports OpenSearch clusters with fine-grained access control only. In the security configuration, the Access policy should look like the image below:



Import an Amazon OpenSearch Service Domain

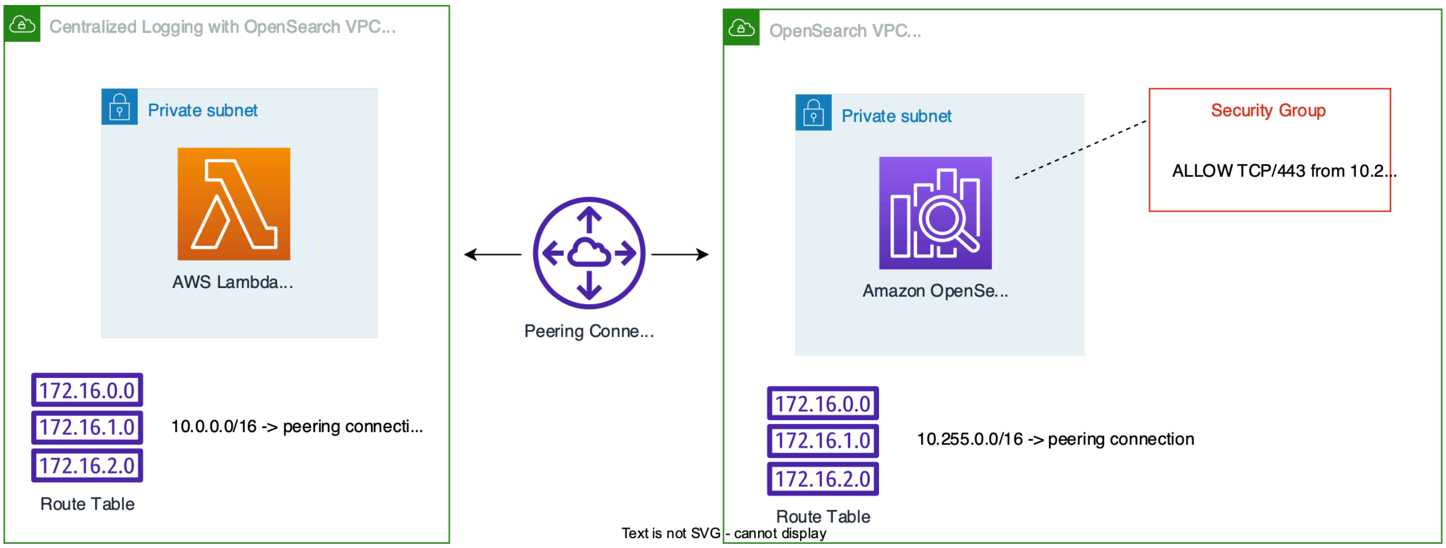
1. Sign in to the Centralized Logging with OpenSearch console.
2. In the left navigation panel, under **Domains**, choose **Import OpenSearch Domain**.
3. On the **Select domain** page, choose a domain from the dropdown list. The dropdown list will display only domains in the same region as the solution.
4. Choose **Next**.
5. On the **Configure network** page, under **Network creation**, choose **Manual** and click **Next**; or choose **Automatic**, and go to step 9.
6. Under **VPC**, choose a VPC from the list. By default, the solution creates a standalone VPC, and you can choose the one named `LogHubVpc/DefaultVPC`. You can also choose the same VPC as your Amazon OpenSearch Service domains.
7. Under **Log Processing Subnet Group**, select at least 2 subnets from the dropdown list. By default, the solution creates two private subnets. You can choose subnets named `LogHubVpc/DefaultVPC/privateSubnet1` and `LogHubVpc/DefaultVPC/privateSubnet2`.
8. Under **Log Processing Security Group**, select one from the dropdown list. By default, the solution creates one Security Group named `ProcessSecurityGroup`.
9. On the **Create tags** page, add tags if needed.
10. Choose **Import**.

Set up VPC Peering

By default, the solution creates a standalone VPC. You need to create VPC Peering to allow the log processing layer to have access to your Amazon OpenSearch Service domains.

Note

Automatic mode will create VPC peering and configure route table automatically. You do not need to set up VPC peering again.



Follow this section to create VPC peering, update security group and update route tables.

Create VPC Peering Connection

1. Sign in to the Centralized Logging with OpenSearch console.
2. In the left navigation panel, under **Domains**, select **OpenSearch Domains**.
3. Find the domain you imported and select the domain name.
4. Choose the **Network** tab.
5. Copy the VPC ID in both sections **OpenSearch domain network** and **Log processing network**. You will create Peering Connection between these two VPCs.
6. Navigate to VPC Console Peering Connections.
7. Select the **Create peering connection** button.
8. On the **Create peering connection** page, enter a name.
9. For the **Select a local VPC to peer with, VPC ID (Requester)**, select the VPC ID of the **Log processing network**.
10. For the **Select another VPC to peer with, VPC ID (Accepter)**, select the VPC ID of the **OpenSearch domain network**.
11. Choose **Create peering connection**, and navigate to the peering connection detail page.

12. Click the **Actions** button and choose **Accept request**.

Update Route Tables

1. Go to the Centralized Logging with OpenSearch console.
2. In the **OpenSearch domain network** section, click the subnet under **AZs and Subnets** to open the subnet console in a new tab.
3. Select the subnet, and choose the **Route table** tab.
4. Select the associated route table of the subnet to open the route table configuration page.
5. Select the **Routes** tab, and choose **Edit routes**.
6. Add a route 10.255.0.0/16 (the CIDR of Centralized Logging with OpenSearch, if you created the solution with existing VPC, please change this value) pointing to the Peering Connection you just created.
7. Go back to the Centralized Logging with OpenSearch console.
8. Click the VPC ID under the **OpenSearch domain network** section.
9. Select the VPC ID on the VPC Console and find its **IPv4 CIDR**.
10. On the Centralized Logging with OpenSearch console, in the **Log processing network** section, click the subnets under **AZs and Subnets** to open the subnets in new tabs.
11. Repeat step 3, 4, 5, 6 to add an opposite route. Namely, configure the IPv4 CIDR of the OpenSearch VPC to point to the Peering Connection. You need to repeat the steps for each subnet of Log processing network.

Update Security Group of OpenSearch Domain

1. On the Centralized Logging with OpenSearch console, under the **OpenSearch domain network** section, select the Security Group ID in **Security Groups** to open the Security Group in a new tab.
2. On the console, select **Edit inbound rules**.
3. Add the rule `ALLOW TCP/443 from 10.255.0.0/16` (the CIDR of Centralized Logging with OpenSearch, if you created Centralized Logging with OpenSearch with existing VPC, change this value).
4. Choose **Save rules**.

Remove an Amazon OpenSearch Service domain

If needed, you can remove the Amazon OpenSearch Service domains.

> **Important**
>
> Removing the domain from Centralized Logging with OpenSearch will **NOT** delete the Amazon OpenSearch Service domain in your AWS account. It will **NOT** impact any existing log analytics pipelines.

1. Sign in to the Centralized Logging with OpenSearch console.
2. In the navigation pane, under **Domains**, choose **OpenSearch Domains**.
3. Select the domain from the table.
4. Choose **Remove**.
5. In the confirmation dialog box, choose **Remove**.

Access proxy

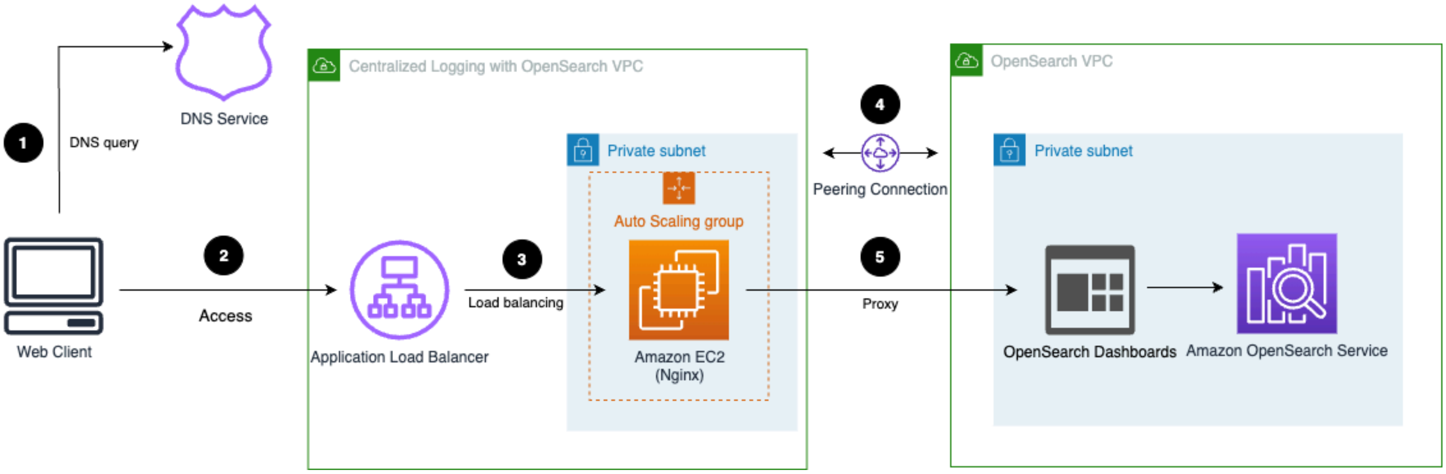
By default, an Amazon OpenSearch Service domain within VPC cannot be accessed from the Internet. Centralized Logging with OpenSearch creates a highly available Nginx cluster which allows you to access the OpenSearch Dashboards from the Internet. Alternatively, you can choose to access the Amazon OpenSearch Service domains using SSH Tunnel.

This section introduces the proxy stack architecture and how to complete the following:

1. Create a proxy
2. Create an associated DNS record
3. Access Amazon OpenSearch Service via proxy
4. Delete a proxy

Architecture

Centralized Logging with OpenSearch creates an Auto Scaling group (ASG) together with an Application Load Balancer (ALB).



The workflow is as follows:

1. Users access the custom domain for the proxy, and the domain needs to be resolved via DNS service (for example, using Route 53 on AWS).
2. The DNS service routes the traffic to internet-facing ALB.
3. The ALB distributes traffic to backend Nginx server running on Amazon EC2 within ASG.
4. The Nginx server redirects the requests to OpenSearch Dashboards.
5. (optional) VPC peering is required if the VPC for the proxy is not the same as the OpenSearch service.

Create a proxy

You can create the Nginx-based proxy using the Centralized Logging with OpenSearch console or by deploying a standalone CloudFormation stack.

Prerequisites

- Make sure an Amazon OpenSearch Service **domain** within VPC is available.
- The domain associated **SSL certificate** is created or uploaded in [Amazon Certificate Manager (ACM)](https://aws.amazon.com/acm/).
- Make sure you have the EC2 private key (.pem) file.

Using the Centralized Logging with OpenSearch console

1. Log in to the Centralized Logging with OpenSearch console.

2. In the navigation pane, under **Domains**, choose **OpenSearch domains**.
3. Select the domain from the table.
4. Under **General configuration**, choose **Enable** at the **Access Proxy** label.

> **Note**
>
> Once the access proxy is enabled, a link to the access proxy will be available.

5. On the **Create access proxy** page, under **Public access proxy**, select at least 2 subnets for **Public Subnets**. You can choose 2 public subnets named LogHubVPC/DefaultVPC/publicSubnet, which are created by Centralized Logging with OpenSearch by default.
6. Choose a Security Group of the ALB in **Public Security Group**. You can choose a security group named ProxySecurityGroup, which is created by Centralized Logging with OpenSearch default.
7. Enter the **Domain Name**.
8. Choose **Load Balancer SSL Certificate** associated with the domain name.
9. Choose the **Nginx Instance Key Name**.
10. Choose **Create**.

Using the CloudFormation stack

This automated AWS CloudFormation template deploys the *Centralized Logging with OpenSearch - Nginx access proxy* solution in the AWS Cloud.

1. Log in to the AWS Management Console and select the button to launch the AWS CloudFormation template.

   Launch Stack

   You can also download the template as a starting point for your own implementation.
2. To launch the stack in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your stack.

5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following parameters.

Parameter	Default	Description
VPCId	*Requires input*	The VPC to deploy the Nginx proxy resources, for example, vpc-bef13dc7.
PublicSubnetIds	*Requires input*	The public subnets where ALB are deployed. You need to select at least two public subnets, for example, subnet-12345abc, subnet-54 321cba.
PrivateSubnetIds	*Requires input*	The private subnets where Nginx instances are deployed. You need to select at least two private subnets, for example, subnet-12 345abc, subnet-54321cba.
NginxSecurityGroupId	*Requires input*	The Security group associate d with the Nginx instances . The security group must allow access from ALB security group.
KeyName	*Requires input*	The PEM key name of the Nginx instances.
ProxyInstanceType	`t3.large`	The OpenSearch proxy instance type.
ProxyInstanceNumber	2	The number of proxy instances.

Parameter	Default	Description
EngineType	OpenSearch	The engine type of the OpenSearch. Select OpenSearch.
Endpoint	*Requires input*	The OpenSearch endpoint, for example, vpc-your_opensearch_domain_name-xcvgw6uu2o6zafsiefxubwuohe.us-east-1.es.amazonaws.com.
CognitoEndpoint	<Optional>	The Cognito User Pool endpoint URL of the OpenSearch domain, for example, mydomain.auth.us-east-1.amazoncognito.com. Leave empty if your OpenSearch domain is not authenticated through Cognito User Pool.
ELBSecurityGroupId	*Requires input*	The Security group being associated with the ALB, for example, sg-123456.
ELBDomain	*Requires input*	The custom domain name of the ALB, for example, dashboard.example.com.
ELBDomainCertificateArn	*Requires input*	The SSL certificate ARN associated with the ELBDomain. The certificate must be created from Amazon Certificate Manager (ACM).

Parameter	Default	Description
ELBAccessLogBucketName	*Requires input*	The Access Log Bucket Name for Proxy ALB.
SsmParameterValueawsserviceamiamazonlinuxlatestamzn2amihvmx8664gp2C96584B6F00A464EAD1953AFF4B05118Parameter	/aws/service/ami-amazon-linux-latest/amzn2-ami-hvm-x86_64-gp2	The SSM parameter of the proxy instance AMI. You can use the default value in most cases.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template creates AWS Identity and Access Management (IAM) resources.
9. Choose **Create** stack to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately 15 minutes.

Recommended Proxy Configuration

The following table provides a list of recommended proxy configuration examples for different number of concurrent users. You can create proxy according to your own use cases.

Number of Concurrent Users	Proxy Instance Type	Number of Proxy Instances
4	t3.nano	1
6	t3.micro	1
8	t3.nano	2
10	t3.small	1
12	t3.micro	2

Number of Concurrent Users	Proxy Instance Type	Number of Proxy Instances
20	t3.small	2
25	t3.large	1
50+	t3.large	2

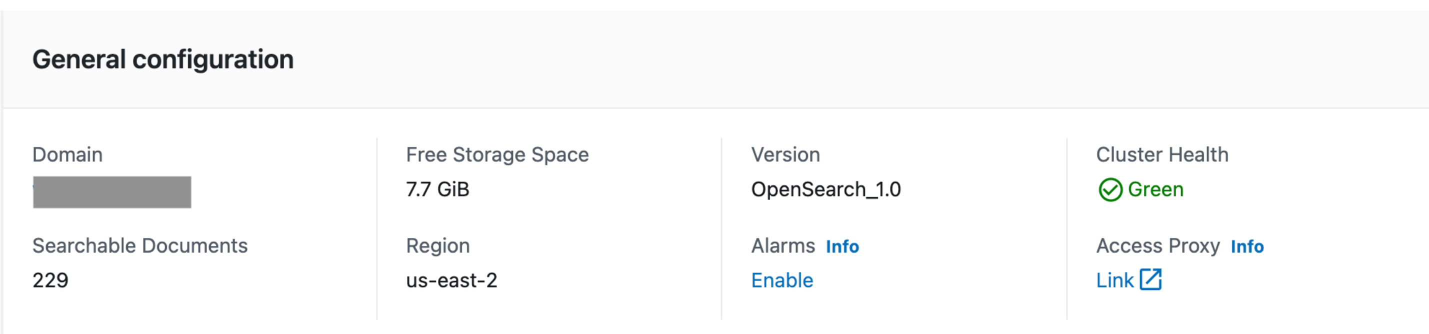
Create an associated DNS record

After provisioning the proxy infrastructure, you need to create an associated DNS record in your DNS resolver. The following introduces how to find the ALB domain, and then create a CNAME record pointing to this domain.

1. Log in to the Centralized Logging with OpenSearch console.
2. In the navigation pane, under **Domains**, choose **OpenSearch domains**.
3. Select the domain from the table.
4. Choose the **Access Proxy** tab. You can see **Load Balancer Domain** which is the ALB domain.
5. Go to the DNS resolver, create a CNAME record pointing to this domain. If your domain is managed by [Amazon Route 53](), refer to [Creating records by using the Amazon Route 53 console]().

Access Amazon OpenSearch Service via proxy

After the DNS record takes effect, you can access the Amazon OpenSearch Service built-in dashboard from anywhere via proxy. You can enter the domain of the proxy in your browser, or click the **Link** button under **Access Proxy** in the **General Configuration** section.



Delete a Proxy

1. Log in to the Centralized Logging with OpenSearch console.

2. In the navigation pane, under **Domains**, choose **OpenSearch domains**.
3. Select the domain from the table.
4. Choose the **Access Proxy** tab.
5. Choose the **Delete**.
6. On the confirmation prompt, choose **Delete**.

Domain Alarms

Amazon OpenSearch Service provides a set of recommended CloudWatch alarms to monitor the health of Amazon OpenSearch Service domains. Centralized Logging with OpenSearch helps you to create the alarms automatically, and send notification to your email (or SMS) via SNS.

Create alarms

Using the Centralized Logging with OpenSearch console

1. Log in to the Centralized Logging with OpenSearch console.
2. In the navigation pane, under **Domains**, choose **OpenSearch domains**.
3. Select the domain from the table.
4. Under **General configuration**, choose **Enable** at the **Alarms** label.
5. Enter the **Email**.
6. Choose the alarms you want to create and adjust the settings if necessary.
7. Choose **Create**.

Using the CloudFormation stack

This automated AWS CloudFormation template deploys the *Centralized Logging with OpenSearch - Alarms* solution in the AWS Cloud.

1. Log in to the AWS Management Console and select the button to launch the AWS CloudFormation template.

   Launch Stack

   You can also download the template as a starting point for your own implementation.
2. To launch the stack in a different AWS Region, use the Region selector in the console navigation bar.

3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your stack.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following parameters.

Parameter	Default	Description
Endpoint	*Requires input*	The endpoint of the OpenSearch domain, for example, vpc-your_opensearch_domain_name-xcvgw6uu2o6zafsiefxubwuohe.us-east-1.es.amazonaws.com.
DomainName	*Requires input*	The name of the OpenSearch domain.
Email	*Requires input*	The notification email address. Alarms will be sent to this email address via SNS.
ClusterStatusRed	Yes	Whether to enable alarm when at least one primary shard and its replicas are not allocated to a node.
ClusterStatusYellow	Yes	Whether to enable alarm when at least one replica shard is not allocated to a node.
FreeStorageSpace	10	Whether to enable alarm when a node in your cluster is down to the free storage space you entered in GiB.

Parameter	Default	Description
		We recommend setting it to 25% of the storage space for each node. 0 means the alarm is disabled.
ClusterIndexWritesBlocked	1	Index writes blocked error occurs for >= x times in 5 minutes, 1 consecutive time. Input 0 to disable this alarm.
UnreachableNodeNumber	3	Nodes minimum is < x for 1 day, 1 consecutive time. 0 means the alarm is disabled.
AutomatedSnapshotFailure	Yes	Whether to enable alarm when automated snapshot failed. AutomatedSnapshotFailure maximum is >= 1 for 1 minute, 1 consecutive time.
CPUUtilization	Yes	Whether to enable alarm when sustained high usage of CPU occurred. CPUUtilization or WarmCPUUtilization maximum is >= 80% for 15 minutes, 3 consecutive times.
JVMMemoryPressure	Yes	Whether to enable alarm when JVM RAM usage peak occurred. JVMMemoryPressure or WarmJVMMemoryPressure maximum is >= 80% for 5 minutes, 3 consecutive times.

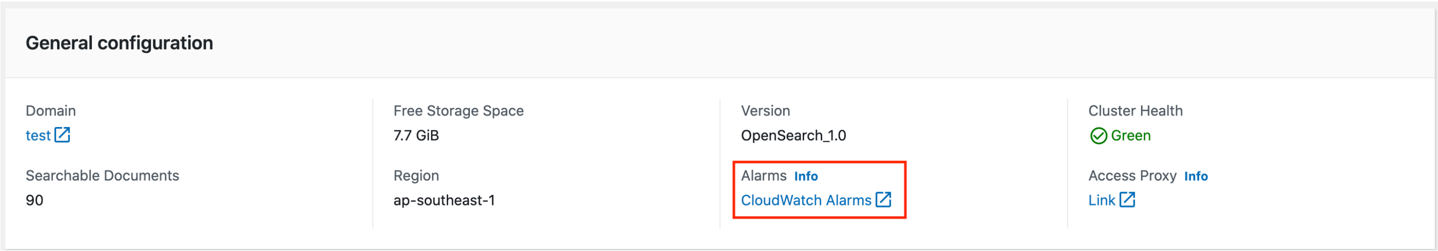
Parameter	Default	Description
MasterCPUUtilization	Yes	Whether to enable alarm when sustained high usage of CPU occurred in master nodes. MasterCPUUtilization maximum is >= 50% for 15 minutes, 3 consecutive times.
MasterJVMMemoryPressure	Yes	Whether to enable alarm when JVM RAM usage peak occurred in master nodes. MasterJVMMemoryPressure maximum is >= 80% for 15 minutes, 1 consecutive time.
KMSKeyError	Yes	Whether to enable alarm when KMS encryption key is disabled. KMSKeyError is >= 1 for 1 minute, 1 consecutive time.
KMSKeyInaccessible	Yes	Whether to enable alarm when KMS encryption key has been deleted or has revoked its grants to OpenSearch Service. KMSKeyInaccessible is >= 1 for 1 minute, 1 consecutive time.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template creates AWS Identity and Access Management (IAM) resources.
9. Choose **Create** stack to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately 5 minutes.

Once you have created the alarms, a confirmation email will be sent to your email address. You need to click the **Confirm** link in the email.

Go to the CloudWatch Alarms page by choosing the **General configuration > Alarms > CloudWatch Alarms** link on the Centralized Logging with OpenSearch console, and the link location is shown as follows:



Make sure that all the alarms are in **OK** status because you might have missed the notification if alarms have changed its status before subscription.

Note

The alarm will not send SNS notification to your email address if triggered before subscription. We recommend you check the alarms status after enabling the OpenSearch alarms. If you see any alarm which is in **In Alarm** status, you should fix that issue first.

Delete alarms

1. Log in to the Centralized Logging with OpenSearch console.
2. In the navigation pane, under **Domains**, choose **OpenSearch domains**.
3. Select the domain from the table.
4. Choose the **Alarms** tab.
5. Choose the **Delete**.
6. On the confirmation prompt, choose **Delete**.

AWS Service Logs

Centralized Logging with OpenSearch supports ingesting AWS service logs into Amazon OpenSearch Service through log analytics pipelines, which you can build using the **Centralized Logging with OpenSearch web console** or via a **standalone CloudFormation template**.

Centralized Logging with OpenSearch reads the data source, parse, cleanup/enrich and ingest logs into Amazon OpenSearch Service domains for analysis. Moreover, the solution provides templated dashboards to facilitate log visualization.

Amazon OpenSearch Service is suitable for real-time log analytics and frequent queries and has full-text search capability.

As of release 2.1.0, the solution starts to support log ingestion into Light Engine, which is suitable for non real-time log analytics and infrequent queries and has SQL-like search capability.

Important

AWS managed services must be in the same region as Centralized Logging with OpenSearch. To ingest logs from different AWS regions, we recommend using S3 cross-region replication. The solution will rotate the index on a daily basis, and cannot be adjusted.

Supported AWS Services

Most of AWS managed services output logs to Amazon CloudWatch Logs, Amazon S3, Amazon Kinesis Data Streams or Amazon Kinesis Firehose.

The following table lists the supported AWS services and the corresponding features.

AWS Service	Log Type	Log Location	Automatic Ingestion	Built-in Dashboard
AWS CloudTrail	N/A	S3	Yes	Yes
Amazon S3	Access logs	S3	Yes	Yes

AWS Service	Log Type	Log Location	Automatic Ingestion	Built-in Dashboard
Amazon RDS/Aurora	MySQL Logs	CloudWatch Logs	Yes	Yes
Amazon CloudFront	Standard access logs	S3	Yes	Yes
Application Load Balancer	Access logs	S3	Yes	Yes
AWS WAF	Web ACL logs	S3	Yes	Yes
AWS Lambda	N/A	CloudWatch Logs	Yes	Yes
Amazon VPC	Flow logs	S3	Yes	Yes
AWS Config	N/A	S3	Yes	Yes

- **Automatic Ingestion**: The solution detects the log location of the resource automatically and then reads the logs.
- **Built-in Dashboard**: An out-of-box dashboard for the specified AWS service. The solution will automatically ingest a dashboard into the Amazon OpenSearch Service.

Most of supported AWS services in Centralized Logging with OpenSearch offers built-in dashboard when creating the log analytics pipelines. You can go to the OpenSearch Dashboards to view the dashboards after the pipeline being provisioned.

In this chapter, you will learn how to create log ingestion and dashboards for the following AWS services:

- AWS CloudTrail
- Amazon S3
- Amazon RDS/Aurora
- Amazon CloudFront
- AWS Lambda

- Application Load Balancer
- AWS WAF
- Amazon VPC
- AWS Config

Cross-Region Logging

When you deploy Centralized Logging with OpenSearch in one Region, the solution allows you to process service logs from another Region.

> **Note**
>
> For Amazon RDS/Aurora and AWS Lambda service logs, this feature is not supported.

The Region where the service resides is referred to as the Source Region. The Region where the Centralized Logging with OpenSearch console is deployed is referred to as the Logging Region.

For AWS CloudTrail, you can create a new trail which send logs into a S3 bucket in the Logging Region. To learn how to create a new trail, please refer to Creating a trail.

For other services with logs located in S3 buckets, you can manually transfer logs (for example, using S3 Cross-Region Replication feature) to the Logging Region S3 bucket.

Follow the steps below to implement Cross-Region Logging:

1. Set the service log location in another Region to be the Logging Region (such as AWS WAF), or automatically copy logs from the Source Region to the Logging Region using CRR.
2. In the solution console, choose **AWS Service Log** in the left navigation pane. Then choose **Create a pipeline**.
3. In the **Select an AWS Service** area, choose a service in the list. Choose **Next**.
4. In **Creation Method**, choose **Manual**, then enter the resource name and S3 log location parameter, and choose **Next**.
5. Set **OpenSearch domain** and **Log Lifecycle** as needed, and choose **Next**.
6. Add tags if you need, and choose **Next** to create the pipeline.

Then you can use the OpenSearch dashboard to discover logs and view dashboards.

AWS CloudTrail Logs

AWS CloudTrail monitors and records account activity across your AWS infrastructure. It outputs all the data to the specified S3 bucket or a CloudWatch log group.

You can create a log ingestion into Amazon OpenSearch Service either by using the Centralized Logging with OpenSearch console or by deploying a standalone CloudFormation stack.

> **Important**
>
> - The CloudTrail logging bucket must be in the same Region as the solution.
> - The Amazon OpenSearch Service index is rotated on a daily basis by default, and you can adjust the index in the Additional Settings.

Create log ingestion (Amazon OpenSearch Service for log analytics)

Using the Centralized Logging with OpenSearch console

1. Sign in to the Centralized Logging with OpenSearch console.
2. In the navigation pane, under **Log Analytics Pipelines**, choose **Service Log**.
3. Choose **Create a log ingestion**.
4. In the **AWS Services** section, choose **AWS CloudTrail**.
5. Choose **Next**.
6. Under **Specify settings**, choose **Automatic** or **Manual**.
   - For **Automatic** mode, choose a CloudTrail from the dropdown list.
   - For **Manual** mode, enter the CloudTrail name.
   - (Optional) If you are ingesting CloudTrail logs from another account, select a linked account from the **Account** dropdown list first.
7. Under **Log Source**, Select **S3** or **CloudWatch** as the log source.
8. Choose **Next**.
9. In the **Specify OpenSearch domain** section, select an imported domain for **Amazon OpenSearch Service domain**.
10. Choose **Yes** for **Sample dashboard** if you want to ingest an associated built-in Amazon OpenSearch Service dashboard.
11. You can change the **Index Prefix** of the target Amazon OpenSearch Service index if needed. The default prefix is your trail name.

12. In the **Log Lifecycle** section, enter the number of days to manage the Amazon OpenSearch Service index lifecycle. Centralized Logging with OpenSearch will create the associated Index State Management (ISM) policy automatically for this pipeline.
13. In the **Select log processor** section, choose the log processor.
    - When selecting Lambda as log processor, you can configure the Lambda concurrency if needed.
    - (Optional) OSI as log processor is now supported in these Regions. When OSI is selected, enter the minimum and maximum number of OCU. For more information, see Scaling pipelines.
14. Choose **Next**.
15. Add tags if needed.
16. Choose **Create**.

Using the CloudFormation Stack

This automated AWS CloudFormation template deploys the *Centralized Logging with OpenSearch - CloudTrail Log Ingestion* solution in the AWS Cloud.

	Launch in AWS Management Console	**Download Template**
AWS Regions	Launch Stack	Template
AWS China Regions	Launch Stack	Template

1. Log in to the AWS Management Console and select above button to launch the AWS CloudFormation template. You can also download the template as a starting point for your own implementation.
2. To launch the stack in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following parameters.

Parameter	Default	Description
Log Bucket Name	*Requires input*	The S3 bucket name which stores the logs.
Log Bucket Prefix	*Requires input*	The S3 bucket path prefix which stores the logs.
Log Source Account ID	<Optional>	The AWS Account ID of the S3 bucket. Required for cross-account log ingestion (Please add a member account first). By default, the Account ID you logged in at **Step 1** will be used.
Log Source Region	<Optional>	The AWS Region of the S3 bucket. By default, the Region you selected at **Step 2** will be used.
Log Source Account Assume Role	<Optional>	The IAM Role ARN used for cross-account log ingestion . Required for cross-account log ingestion (Please add a member account first).
Engine Type	OpenSearch	The engine type of the OpenSearch. Select OpenSearch.
OpenSearch Domain Name	*Requires input*	The domain name of the Amazon OpenSearch Service cluster.

Parameter	Default	Description
OpenSearch Endpoint	*Requires input*	The OpenSearch endpoint URL. For example, vpc-your_opensearch_domain_name-xcvgw6uu2o6zafsiefxubwuohe.us-east-1.es.amazonaws.com
Index Prefix	*Requires input*	The common prefix of OpenSearch index for the log. The index name will be <Index Prefix>-<Log Type>-<Other Suffix>.
Create Sample Dashboard	Yes	Whether to create a sample OpenSearch dashboard.
VPC ID	*Requires input*	Select a VPC which has access to the OpenSearch domain. The log processing Lambda will reside in the selected VPC.
Subnet IDs	*Requires input*	Select at least two subnets which have access to the OpenSearch domain. The log processing Lambda will reside in the subnets. Make sure the subnets have access to the Amazon S3 service.
Security Group ID	*Requires input*	Select a Security Group which will be associated with the log processing Lambda. Make sure the Security Group has access to the OpenSearch domain.

Parameter	Default	Description
S3 Backup Bucket	`Requires input`	The S3 backup bucket name to store the failed ingestion logs.
KMS-CMK ARN	<Optional>	The KMS-CMK ARN for encryption. Leave it blank to create a new KMS CMK.
Number Of Shards	5	Number of shards to distribute the index evenly across all data nodes. Keep the size of each shard between 10-50 GB.
Number of Replicas	1	Number of replicas for OpenSearch Index. Each replica is a full copy of an index.
Age to Warm Storage	<Optional>	The age required to move the index into warm storage (e.g. 7d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). This is only effective when warm storage is enabled in OpenSearch.

Parameter	Default	Description
Age to Cold Storage	<Optional>	The age required to move the index into cold storage (e.g. 30d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). This is only effective when cold storage is enabled in OpenSearch.
Age to Retain	<Optional>	The age to retain the index (e.g. 180d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). If value is "", the index will not be deleted.
Rollover Index Size	<Optional>	The minimum size of the shard storage required to roll over the index (e.g. 30GB).
Index Suffix	yyyy-MM-dd	The common suffix format of OpenSearch index for the log(Example: yyyy-MM-dd, yyyy-MM-dd-HH). The index name will be <Index Prefix>-<Log Type>-<Index Suffix>-000001.
Compression type	best_compression	The compression type to use to compress stored data. Available values are best_compression and default.

Parameter	Default	Description
Refresh Interval	1s	How often the index should refresh, which publishes its most recent changes and makes them available for searching. Can be set to -1 to disable refreshing. Default is 1s.
EnableS3Notification	True	An option to enable or disable notifications for Amazon S3 buckets. The default option is recommend ed for most cases.
LogProcessorRoleName	<Optional>	Specify a role name for the log processor. The name should NOT duplicate an existing role name. If no name is specified, a random name is generated.
QueueName	<Optional>	Specify a queue name for an SQS. The name should NOT duplicate an existing queue name. If no name is given, a random name is generated.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template creates AWS Identity and Access Management (IAM) resources.
9. Choose **Create** stack to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately 10 minutes.

View dashboard

The dashboard includes the following visualizations.

Visualization Name	Source Field	Description
Global Control	awsRegion	Provides users with the ability to drill down data by Region.
Event History	log event	Presents a bar chart that displays the distribution of events over time.
Event by Account ID	userIdentity.accountId	Breaks down events based on the AWS account ID, enabling you to analyze activity patterns across different accounts within your organization.
Top Event Names	eventName	Shows the most frequentl y occurring event names, helping you identify common activities or potential anomalies.
Top Event Sources	eventSource	Highlights the top sources generating events, providing insights into the services or resources that are most active or experiencing the highest event volume.
Event Category	eventCategory	Categorizes events into different types or classific ations, facilitating analysis and understanding of event distribution across categories.

Visualization Name	Source Field	Description
Top Users	• userIdentity.sessionContext .sessionIssuer.userName • userIdentity.sessionContext .sessionIssuer.arn • userIdentity.accountId • userIdentity.sessionContext .sessionIssuer.type	Identifies the users or IAM roles associated with the highest number of events, aiding in user activity monitoring and access management.
Top Source IPs	sourceIPAddress	Lists the source IP addresses associated with events, enabling you to identify and investigate potentially suspicious or unauthorized activities.
S3 Access Denied	• eventSource: s3* • errorCode: AccessDenied	Displays events where access to Amazon S3 resources was denied, helping you identify and troubleshoot permissio n issues or potential security breaches.
S3 Buckets	requestParameters. bucketName	Provides a summary of S3 bucket activity, including create, delete, and modify operations, allowing you to monitor changes and access patterns.
Top S3 Change Events	• eventName • requestParameters. bucketName	Presents the most common types of changes made to S3 resources, such as object uploads, deletions, or modifications, aiding in change tracking and auditing.

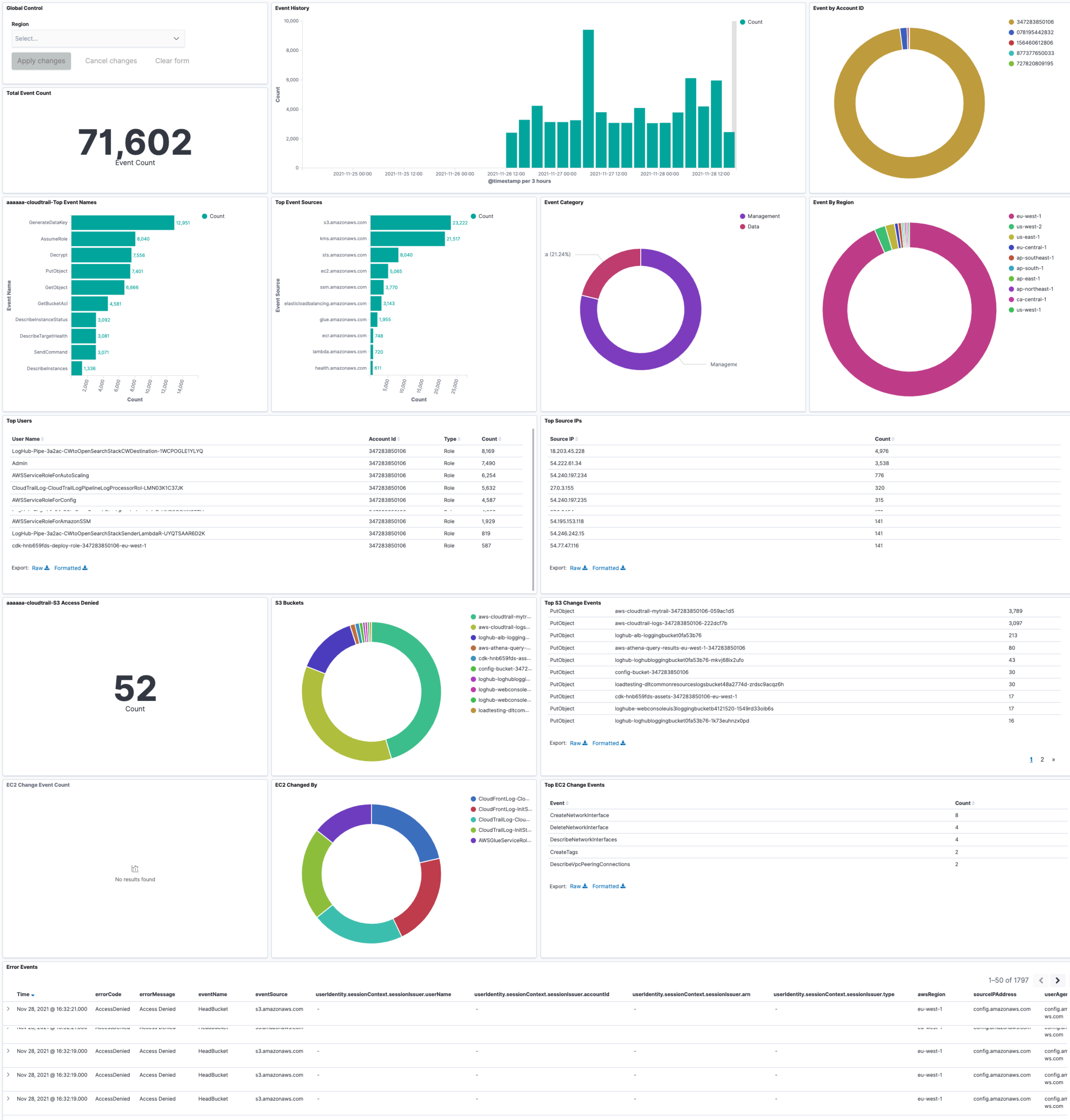
Visualization Name	Source Field	Description
EC2 Change Event Count	• eventSource: ec2* • eventName: (RunInstances or TerminateInstances or RunInstances or StopInsta nces)	Shows the total count of EC2-related change events, giving an overview of the volume and frequency of changes made to EC2 instances and resources.
EC2 Changed By	userIdentity.sessionContext .sessionIssuer.userName	Identifies the users or IAM roles responsible for changes to EC2 resources, assisting in accountability and tracking of modifications.
Top EC2 Change Events	eventName	Highlights the most common types of changes made to EC2 instances or related resources , allowing you to focus on the most significant or frequent changes.

Visualization Name	Source Field	Description
Error Events	• awsRegion • errorCode • errorMessage • eventName • eventSource • sourceIPAddress • userAgent • userIdentity.accountId • userIdentity.sessionContext.sessionIssuer.accountId • userIdentity.sessionContext.sessionIssuer.arn • userIdentity.sessionContext.sessionIssuer.type • userIdentity.sessionContext.sessionIssuer.userName	Displays events that resulted in errors or failures, helping you identify and troubleshoot issues related to API calls or resource operations.

Sample dashboard

You can access the built-in dashboard in Amazon OpenSearch Service to view log data. For more information, see Access Dashboard.

You can click the below image to view the high-resolution sample dashboard.



Create log ingestion (Light Engine for log analytics)

Using the Centralized Logging with OpenSearch Console

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the navigation pane, under **Log Analytics Pipelines**, choose **Service Log**.
3. Choose the **Create a log ingestion** button.
4. In the **AWS Services** section, choose **AWS CloudTrail**.
5. Choose **Light Engine**, and choose **Next**.
6. Under **Specify settings**, choose **Automatic** or **Manual** for **CloudTrail logs enabling**. The automatic mode will detect the CloudTrail log location automatically.
    - For **Automatic mode**, choose the CloudTrail from the dropdown list.
    - For Standard Log, the solution will automatically detect the log location if logging is enabled.
    - For **Manual mode**, enter the **CloudTrail ID** and **CloudTrail Standard Log location**.
    - (Optional) If you are ingesting CloudFront logs from another account, select a linked account from the **Account** dropdown list first.
7. Choose **Next**.
8. In the **Specify Light Engine Configuration** section, if you want to ingest an associated templated Grafana dashboard, select **Yes** for the sample dashboard.
9. Choose an existing Grafana, or import a new one by making configurations in Grafana.
10. Select an Amazon S3 bucket to store partitioned logs and give a name to the log table. The solution provides a predefined table name, but you can modify it according to your needs.
11. Modify the log processing frequency if needed, which is set to **5** minutes by default with a minimum processing frequency of **1** minute.
12. In the **Log Lifecycle** section, if needed, enter the log merge time and log archive time to modify the default values provided by the solution.
13. Choose **Next**.
14. Add tags if needed.
15. Choose **Create**.

Using the CloudFormation Stack

This automated AWS CloudFormation template deploys the *Centralized Logging with OpenSearch - CloudTrail Standard Log Ingestion* template in the AWS Cloud.

	Launch in AWS Management Console	Download Template
AWS Regions	Launch Stack	Template
AWS China Regions	Launch Stack	Template

1. Log in to the AWS Management Console and select the button to launch the AWS CloudFormation template. You can also download the template as a starting point for your own implementation.
2. To launch the stack in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following parameters.
   - Parameters for **Pipeline settings**

Parameter	Default	Description
Pipeline Id	`Requires input`	The unique identifier for the pipeline, which is essential if you need to create multiple ALB pipelines and write different ALB logs into separate tables. To ensure uniqueness, you can generate a unique pipeline identifier using uuidgenerator.

Parameter	Default	Description
Staging Bucket Prefix	AWSLogs/CloudTrailLogs	The storage directory for logs in the temporary storage area should ensure uniqueness and non-overlapping of the prefix for different pipelines.

- Parameters for **Destination settings**

Parameter	Default	Description
Centralized Bucket Name	*`Requires input`*	The name for the centralized S3 bucket. For example, `centralized-logging-bucket` .
Centralized Bucket Prefix	datalake	The centralized bucket prefix. By default, the database location is `s3://{Centralized Bucket Name}/{Centralized Bucket Prefix}/amazon_cl_centralized` .
Centralized Table Name	CloudTrail	Table name for writing data to the centralized database. You can modify it if needed.

- Parameters for **Scheduler settings**

Parameter	Default	Description
LogProcessor Schedule Expression	rate(5 minutes)	Task scheduling expression for performing log processin g, with a default value of executing the LogProcessor every 5 minutes. For more information, see Schedule types.
LogMerger Schedule Expression	cron(0 1 * * ? *)	Task scheduling expression for performing log merging, with a default value of executing the LogMerger at 1 AM every day. For more information, see Schedule types.
LogArchive Schedule Expression	cron(0 2 * * ? *)	Task scheduling expression for performing log archiving , with a default value of executing the LogArchive at 2 AM every day. For more information, see Schedule types.
Age to Merge	7	Small file retention days, with a default value of 7, indicating that logs older than 7 days will be merged into small files. It can be adjusted as needed.

Parameter	Default	Description
Age to Archive	30	Log retention days, with a default value of 30, indicating that data older than 30 days will be archived and deleted. It can be adjusted as needed.

- Parameters for **Notification settings**

Parameter	Default	Description
Notification Service	SNS	Notification method for alerts. If your main stack is in AWS China Regions, you can only choose the SNS method. If your main stack is in AWS Regions, you can choose either the SNS or SES method.

Parameter	Default	Description
Recipients	`Requires input`	If the Notification Service is SNS, enter the SNS Topic ARN to ensure that you have the required permissions. If the Notification Service is SES, enter the email addresses separated by commas to ensure that the email addresses are already Verified Identities in SES. The adminEmail provided during the creation of the main stack will receive a verification email by default.

- Parameters for **Dashboard settings**

Parameter	Default	Description
Import Dashboards	FALSE	Whether to import the Dashboard into Grafana. By default, it is `false`. If it is set to `true`, you must provide the Grafana URL and Grafana Service Account Token.
Grafana URL	`Requires input`	Grafana access URL. For example, `https://alb-72277319.us-west-2.elb.amazonaws.com` .

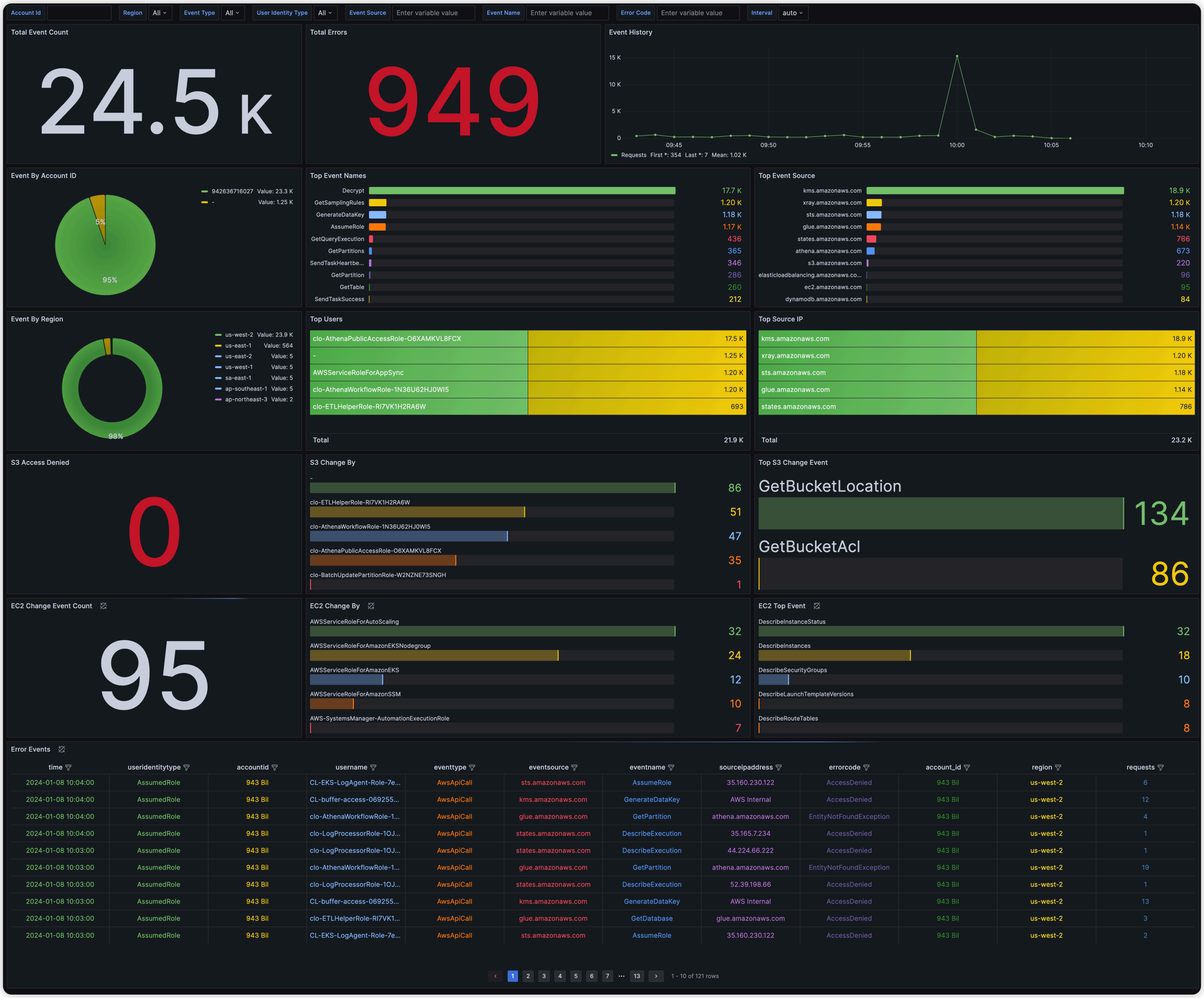
Parameter	Default	Description
Grafana Service Account Token	*Requires input*	Service Account Token created in Grafana.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template creates AWS Identity and Access Management (IAM) resources.
9. Choose **Create** stack to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately 10 minutes.

Sample dashboard

Below shows the sample dashboard.



Amazon S3 Logs

[Amazon S3 server access logging](#) provides detailed records for the requests made to the bucket. S3 access logs can be enabled and saved in another S3 bucket.

Create log ingestion

You can create a log ingestion into Amazon OpenSearch Service either by using the Centralized Logging with OpenSearch console or by deploying a standalone CloudFormation stack.

Important

- The S3 Bucket region must be the same as the Centralized Logging with OpenSearch solution region.
- The Amazon OpenSearch Service index is rotated on a daily basis by default, and you can adjust the index in the Additional Settings.

Using the Centralized Logging with OpenSearch Console

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the navigation pane, under **Log Analytics Pipelines**, choose **Service Log**.
3. Choose the **Create a log ingestion** button.
4. In the **AWS Services** section, choose **Amazon S3**.
5. Choose **Next**.
6. Under **Specify settings**, choose **Automatic** or **Manual** for **S3 Access Log enabling**. The automatic mode will enable the S3 Access Log and save the logs to a centralized S3 bucket if logging is not enabled yet.
   - For **Automatic mode**, choose the S3 bucket from the dropdown list.
   - For **Manual mode**, enter the **Bucket Name** and **S3 Access Log location**.
   - (Optional) If you are ingesting Amazon S3 logs from another account, select a linked account from the **Account** dropdown list first.
7. Choose **Next**.
8. In the **Specify OpenSearch domain** section, select an imported domain for **Amazon OpenSearch Service domain**.
9. Choose **Yes** for **Sample dashboard** if you want to ingest an associated built-in Amazon OpenSearch Service dashboard.
10. You can change the **Index Prefix** of the target Amazon OpenSearch Service index if needed. The default prefix is your bucket name.
11. In the **Log Lifecycle** section, enter the number of days to manage the Amazon OpenSearch Service index lifecycle. The Centralized Logging with OpenSearch will create the associated Index State Management (ISM) policy automatically for this pipeline.
12. Choose **Next**.
13. Add tags if needed.
14. Choose **Create**.

Using the standalone CloudFormation Stack

This automated AWS CloudFormation template deploys the *Centralized Logging with OpenSearch - S3 Access Log Ingestion* solution in the AWS Cloud.

	Launch in AWS Management Console	Download Template
AWS Regions	Launch Stack	[Template](#)
AWS China Regions	Launch Stack	[Template](#)

1. Log in to the AWS Management Console and select above button to launch the AWS CloudFormation template. You can also download the template as a starting point for your own implementation.
2. To launch the stack in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following parameters.

Parameter	Default	Description
Log Bucket Name	`Requires input`	The S3 bucket name which stores the logs.
Log Bucket Prefix	`Requires input`	The S3 bucket path prefix which stores the logs.
Log Source Account ID	<Optional>	The AWS Account ID of the S3 bucket. Required for cross-account log ingestion (Please add a member

Parameter	Default	Description
		account first). By default, the Account ID you logged in at **Step 1** will be used.
Log Source Region	<Optional>	The AWS Region of the S3 bucket. By default, the Region you selected at **Step 2** will be used.
Log Source Account Assume Role	<Optional>	The IAM Role ARN used for cross-account log ingestion . Required for cross-account log ingestion (Please add a member account first).
Engine Type	OpenSearch	The engine type of the OpenSearch. Select OpenSearch.
OpenSearch Domain Name	*Requires input*	The domain name of the Amazon OpenSearch Service cluster.
OpenSearch Endpoint	*Requires input*	The OpenSearch endpoint URL. For example, `vpc-your_opensearch_domain_name-xcvgw6uu2o6zafsiefxubwuohe.us-east-1.es.amazonaws.com`
Index Prefix	*Requires input*	The common prefix of OpenSearch index for the log. The index name will be `<Index Prefix>-<Log Type>-<Other Suffix>` .

Parameter	Default	Description
Create Sample Dashboard	Yes	Whether to create a sample OpenSearch dashboard.
VPC ID	*Requires input*	Select a VPC which has access to the OpenSearch domain. The log processor Lambda function will reside in the selected VPC.
Subnet IDs	*Requires input*	Select at least two subnets which have access to the OpenSearch domain. The log processor Lambda function will reside in the subnets. Make sure the subnets have access to the Amazon S3 service.
Security Group ID	*Requires input*	Select a Security Group which will be associated with the log processor Lambda function. Make sure the Security Group has access to the OpenSearch domain.
S3 Backup Bucket	*Requires input*	The S3 backup bucket name to store the failed ingestion logs.
KMS-CMK ARN	<Optional>	The KMS-CMK ARN for encryption. Leave it blank to create a new KMS CMK.

Parameter	Default	Description
Number Of Shards	5	Number of shards to distribute the index evenly across all data nodes. Keep the size of each shard between 10-50 GB.
Number of Replicas	1	Number of replicas for OpenSearch Index. Each replica is a full copy of an index.
Age to Warm Storage	<Optional>	The age required to move the index into warm storage (e.g. 7d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). This is only effective when warm storage is enabled in OpenSearch.
Age to Cold Storage	<Optional>	The age required to move the index into cold storage (e.g. 30d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). This is only effective when cold storage is enabled in OpenSearch.

Parameter	Default	Description
Age to Retain	<Optional>	The age to retain the index (e.g. 180d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). If value is "", the index will not be deleted.
Rollover Index Size	<Optional>	The minimum size of the shard storage required to roll over the index (e.g. 30GB).
Index Suffix	yyyy-MM-dd	The common suffix format of OpenSearch index for the log(Example: yyyy-MM-dd, yyyy-MM-dd-HH). The index name will be `<Index Prefix>-<Log Type>-<Index Suffix>-000001` .
Compression type	best_compression	The compression type to use to compress stored data. Available values are best_compression and default.
Refresh Interval	1s	How often the index should refresh, which publishes its most recent changes and makes them available for searching. Can be set to -1 to disable refreshing. Default is 1s.

Parameter	Default	Description
EnableS3Notification	True	An option to enable or disable notifications for Amazon S3 buckets. The default option is recommend ed for most cases.
LogProcessorRoleName	<Optional>	Specify a role name for the log processor. The name should NOT duplicate an existing role name. If no name is specified, a random name is generated.
QueueName	<Optional>	Specify a queue name for an SQS. The name should NOT duplicate an existing queue name. If no name is given, a random name is generated.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template creates AWS Identity and Access Management (IAM) resources.
9. Choose **Create** stack to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately 10 minutes.

View dashboard

The dashboard includes the following visualizations.

Visualization Name	Source Field	Description
Total Requests	• log event	A visualization showing the total number of requests made to the S3 bucket, including all types of operations (e.g., GET, PUT, DELETE).
Unique Visitors	• log event	This visualization displays the count of unique visitors accessing the S3 bucket, identified by their IP addresses.
Access History	• log event	Provides a chronological log of all access events made to the S3 bucket, including details about the operations and their outcomes.
Request By Operation	• operation	This visualization categorizes and shows the distribution of requests based on different operations, such as GET, PUT, DELETE, etc.
Status Code	• http_status	Displays the count of requests made to the S3 bucket, grouped by HTTP status codes returned by the server (e.g., 200, 404, 403, etc.).
Status Code History	• http_status	Shows the historical trend of HTTP status codes returned by the Amazon S3 server over a specific period of time.

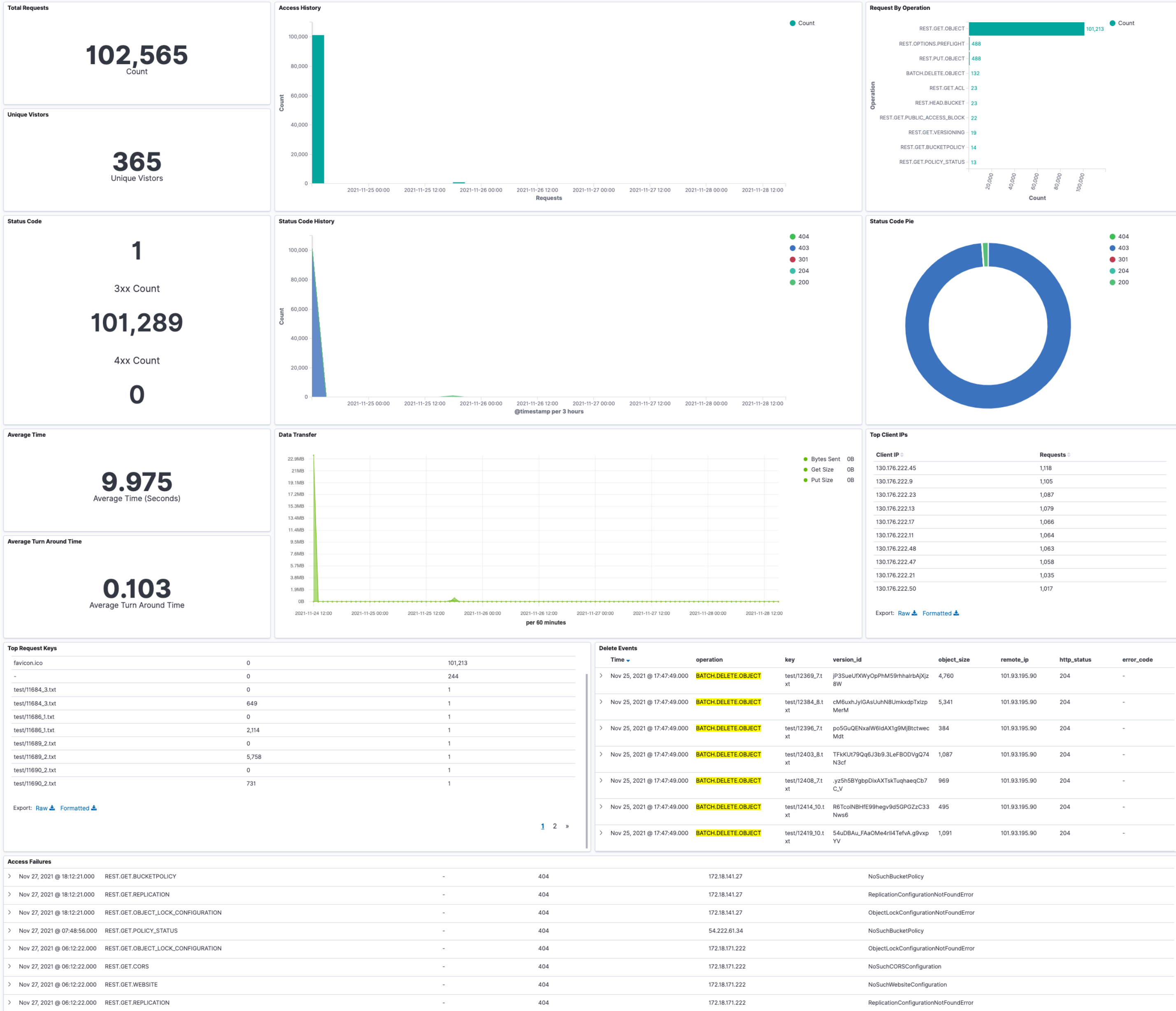
Visualization Name	Source Field	Description
Status Code Pie	• http_status	Represents the distribution of requests based on different HTTP status codes using a pie chart.
Average Time	• total_time	This visualization calculate s and presents the average time taken for various operations in the S3 bucket (e.g., average time for GET, PUT requests, etc.).
Average Turn Around Time	• turn_around_time	Shows the average turnaroun d time for different operation s, which is the time between receiving a request and sending the response back to the client.
Data Transfer	• bytes_sent • object_size • operation	Provides insights into data transfer activities, including the total bytes transferred, object sizes, and different operations involved.
Top Client IPs	• remote_ip	Displays the top client IP addresses with the highest number of requests made to the S3 bucket.
Top Request Keys	• key • object_size	Shows the top requested keys in the S3 bucket along with the corresponding object sizes.

Visualization Name	Source Field	Description
Delete Events	• operation • key • version_id • object_size • remote_ip • http_status • error_code	Focuses on delete events, including the operation, key, version ID, object size, client IP, HTTP status, and error code associated with the delete requests.
Access Failures	• operation • key • version_id • object_size • remote_ip • http_status • error_code	Highlights access failures, showing the details of the failed requests, including operation, key, version ID, object size, client IP, HTTP status, and error code.

Sample Dashboard

You can access the built-in dashboard in Amazon OpenSearch Service to view log data. For more information, see Access Dashboard.

You can click the below image to view the high-resolution sample dashboard.



Amazon RDS/Aurora Logs

You can [publish database instance logs to Amazon CloudWatch Logs](). Then, you can perform real-time analysis of the log data, store the data in highly durable storage, and manage the data with the CloudWatch Logs Agent.

Prerequisites

Make sure your database logs are enabled. Some databases logs are not enabled by default, and you need to update your database parameters to enable the logs.

Refer to How do I enable and monitor logs for an Amazon RDS MySQL DB instance? to learn how to output logs to CloudWatch Logs.

The table below lists the requirements for Amazon RDS/Aurora MySQL parameters.

Parameter	Requirement
Audit Log	The database instance must use a custom option group with the MARIADB_AUDIT_PLUG IN option.
General log	The database instance must use a custom parameter group with the parameter setting general_log = 1 to enable the general log.
Slow query log	The database instance must use a custom parameter group with the parameter setting slow_query_log = 1 to enable the slow query log.
Log output	The database instance must use a custom parameter group with the parameter setting log_output = FILE to write logs to the file system and publish them to CloudWatch Logs.

Create log ingestion

You can create a log ingestion into Amazon OpenSearch Service either by using the Centralized Logging with OpenSearch console or by deploying a standalone CloudFormation stack.

Important

The RDS and CloudWatch region must be the same as the Centralized Logging with OpenSearch solution region.

The Amazon OpenSearch Service index is rotated on a daily basis by default, and you can adjust the index in the Additional Settings.

Using the Centralized Logging with OpenSearch Console

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the navigation pane, under **Log Analytics Pipelines**, choose **Service Log**.
3. Choose the **Create a log ingestion** button.
4. In the **AWS Services** section, choose **Amazon RDS**.
5. Choose **Next**.
6. Under **Specify settings**, choose **Automatic** or **Manual** for **RDS log enabling**. The automatic mode will detect your RDS log configurations and ingest logs from CloudWatch.
   - For **Automatic mode**, choose the RDS cluster from the dropdown list.
   - For **Manual mode**, enter the **DB identifier**, select the **Database type** and input the CloudWatch log location in **Log type and location**.
   - (Optional) If you are ingesting RDS/Aurora logs from another account, select a linked account from the **Account** dropdown first.
7. Choose **Next**.
8. In the **Specify OpenSearch domain** section, select an imported domain for **Amazon OpenSearch Service domain**.
9. Choose **Yes** for **Sample dashboard** if you want to ingest an associated templated Amazon OpenSearch Service dashboard.
10. You can change the **Index Prefix** of the target Amazon OpenSearch Service index if needed. The default prefix is the Database identifier.
11. In the **Log Lifecycle** section, input the number of days to manage the Amazon OpenSearch Service index lifecycle. The Centralized Logging with OpenSearch will create the associated Index State Management (ISM) policy automatically for this pipeline.
12. Choose **Next**.
13. Add tags if needed.
14. Choose **Create**.

Using the CloudFormation Stack

This automated AWS CloudFormation template deploys the *Centralized Logging with OpenSearch - RDS Log Ingestion* solution in the AWS Cloud.

	Launch in AWS Management Console	Download Template
AWS Regions	Launch Stack	Template
AWS China Regions	Launch Stack	Template

1. Log in to the AWS Management Console and select the button to launch the AWS CloudFormation template. You can also download the template as a starting point for your own implementation.
2. To launch the Centralized Logging with OpenSearch in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following parameters.

Parameter	Default	Description
Log Bucket Name	`Requires input`	The S3 bucket name to export the logs.
Log Bucket Prefix	`Requires input`	The S3 bucket path prefix which stores the the logs.
Log Source Account ID	<Optional>	The AWS Account ID of the CloudWatch log group. Required for cross-account log ingestion (Please add a member account first). By default, the Account ID you logged in at **Step 1** will be used.

Parameter	Default	Description
Log Source Region	<Optional input>	The AWS Region of the CloudWatch log group. By default, the Region you selected at **Step 2** will be used.
Log Source Account Assume Role	<Optional input>	The IAM Role ARN used for cross-account log ingestion . Required for cross-account log ingestion (Please add a member account first).
Log Group Names	*`Requires input`*	The names of the CloudWatch log group for the logs.
Engine Type	OpenSearch	The engine type of the OpenSearch. Select OpenSearch.
OpenSearch Domain Name	*`Requires input`*	The domain name of the Amazon OpenSearch Service cluster.
OpenSearch Endpoint	*`Requires input`*	The OpenSearch endpoint URL. For example, vpc-your_opensearch_domain_name-xcvgw6uu2o6zafsiefxubwuohe.us-east-1.es.amazonaws.com
Index Prefix	*`Requires input`*	The common prefix of OpenSearch index for the log. The index name will be <Index Prefix>-<Log Type>-<Other Suffix>.

Parameter	Default	Description
Create Sample Dashboard	Yes	Whether to create a sample OpenSearch dashboard.
VPC ID	*Requires input*	Select a VPC which has access to the OpenSearch domain. The log processor Lambda function will reside in the selected VPC.
Subnet IDs	*Requires input*	Select at least two subnets which has access to the OpenSearch domain. The log processor Lambda function will reside in the subnets. Please make sure the subnets has access to the Amazon S3 service.
Security Group ID	*Requires input*	Select a Security Group which will be associated to the log processor Lambda function. Please make sure the Security Group has access to the OpenSearch domain.
S3 Backup Bucket	*Requires input*	The S3 backup bucket name to store the failed ingestion logs.
KMS-CMK ARN	<Optional input>	The KMS-CMK ARN for SQS encryption. Leave it blank to create a new KMS CMK.

Parameter	Default	Description
Number Of Shards	5	Number of shards to distribute the index evenly across all data nodes. Keep the size of each shard between 10-50 GiB.
Number of Replicas	1	Number of replicas for OpenSearch Index. Each replica is a full copy of an index.
Age to Warm Storage	<Optional>	The age required to move the index into warm storage (e.g. 7d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). This is only effective when warm storage is enabled in OpenSearch.
Age to Cold Storage	<Optional>	The age required to move the index into cold storage (e.g. 30d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). This is only effective when cold storage is enabled in OpenSearch.

Parameter	Default	Description
Age to Retain	<Optional>	The age to retain the index (e.g. 180d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). If value is "", the index will not be deleted.
Rollover Index Size	<Optional>	The minimum size of the shard storage required to roll over the index (e.g. 30GB).
Index Suffix	yyyy-MM-dd	The common suffix format of OpenSearch index for the log (Example: yyyy-MM-dd, yyyy-MM-dd-HH). The index name will be <Index Prefix>-<Log Type>-<Index Suffix>-000001.
Compression type	best_compression	The compression type to use to compress stored data. Available values are best_compression and default.
Refresh Interval	1s	How often the index should refresh, which publishes its most recent changes and makes them available for searching. Can be set to -1 to disable refreshing. Default is 1s.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.

8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template creates AWS Identity and Access Management (IAM) resources.
9. Choose **Create** stack to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately 15 minutes.

View dashboard

The dashboard includes the following visualizations.

Visualization Name	Source Field	Description
Controller	• db-identifier • sq-table-name	This visualization allows users to filter data based on the db-identifier and sq-table-name fields.
Total Log Events Overview	• db-identifier • log event	This visualization presents an overview of the total log events for the specified database ('db-identifier'). It helps monitor the frequency of various log events.
Slow Query History	• log event	This visualization shows the historical data of slow query log events. It allows you to track the occurrences of slow queries and identify potential performance issues.
Average Slow Query Time History	• Average sq-duration	This visualization depicts the historical trend of the average duration of slow queries ('sq-duration'). It helps in understanding the database' s performance over time and

Visualization Name	Source Field	Description
		identifying trends related to slow query durations.
Total Slow Queries	• log event	This visualization provides the total count of slow queries in the log events. It gives an immediate view of how many slow queries have occurred during a specific time period, which is useful for assessing the database's performance and potential bottlenecks.
Average Slow Query Duration	• Average sq-duration	This visualization shows the average duration of slow queries ('sq-duration') over time. It is valuable for understanding the typical performance of slow queries in the database.
Top Slow Query IP	• sq-ip • sq-duration	This visualization highlight s the IP addresses ('sq-ip') associated with the slowest queries and their respective durations ('sq-duration'). It helps identify sources of slow queries and potential areas for optimization.

Visualization Name	Source Field	Description
Slow Query Scatter Plot	• sq-duration • sq-ip • sq-query	This scatter plot visualization represents the relationship between the duration of slow queries ('sq-duration'), the IP addresses ('sq-ip') from which they originated, and the actual query content ('sq-quer y'). It helps in understanding query performance patterns and identifying potential issues related to specific queries and their sources.
Slow Query Pie	• sq-query	This pie chart visualiza tion shows the distribution of slow queries based on their content ('sq-query'). It provides an overview of the types of queries causing performance issues, allowing you to focus on optimizing specific query patterns.
Slow Query Table Name Pie	• sq-table-name	This pie chart visualization displays the distribution of slow queries based on the table names ('sq-table-name') they access. It helps identify which tables are affected by slow queries, enabling targeted optimization efforts for specific tables.

Visualization Name	Source Field	Description
Top Slow Query	• sq-query	This visualization presents the slowest individual queries based on their content ('sq-query'). It is helpful in pinpointing specific queries that have the most significa nt impact on performance, allowing developers and administrators to focus on optimizing these critical queries.
Slow Query Logs	• db-identifier • sq-db-name • sq-table-name • sq-query • sq-ip • sq-host-name • sq-rows-examined • sq-rows-sent • sq-id • sq-duration • sq-lock-wait	This visualization provides detailed logs of slow queries, including database ('sq-db-name'), table ('sq-table-name'), query content ('sq-query'), IP address ('sq-ip'), host name ('sq-host-name'), rows examined ('sq-rows-examined'), rows sent ('sq-rows-sent'), query ID ('sq-id'), query duration ('sq-duration'), and lock wait time ('sq-lock-wait'). It is beneficia l for in-depth analysis and troubleshooting of slow query performance.

Visualization Name	Source Field	Description
Total Deadlock Queries	• log event	This visualization shows the total number of deadlock occurrences based on the log events. Deadlocks are critical issues that can cause database transactions to fail, and monitoring their frequency is essential for ensuring database stability.
Deadlock History	• log event	This visualization displays the historical data of deadlock occurrences based on the log events. Understanding the pattern of deadlocks over time can help identify recurring issues and take preventive measures to reduce their impact on the database.

Visualization Name	Source Field	Description
Deadlock Query Logs	• db-identifier • log-detail • deadlock-ip-1 • deadlock-action-1 • deadlock-os-thread-handle-1 • deadlock-query-1 • deadlock-query-id-1 • deadlock-thread-id-1 • deadlock-user-1 • deadlock-action-2 • deadlock-ip-2 • deadlock-os-thread-handle-2 • deadlock-query-2 • deadlock-query-id-2 • deadlock-thread-id-2 • deadlock-user-2	This visualization provides detailed logs of deadlock occurrences
Total Error Logs	• log event	This visualization presents the total count of error log events. Monitoring error logs helps identify database issues and potential errors that need attention and resolution.
Error History	• log event	This visualization shows the historical data of error log events. Understanding the error patterns over time can aid in identifying recurring issues and taking corrective actions to improve the database's overall health and stability.

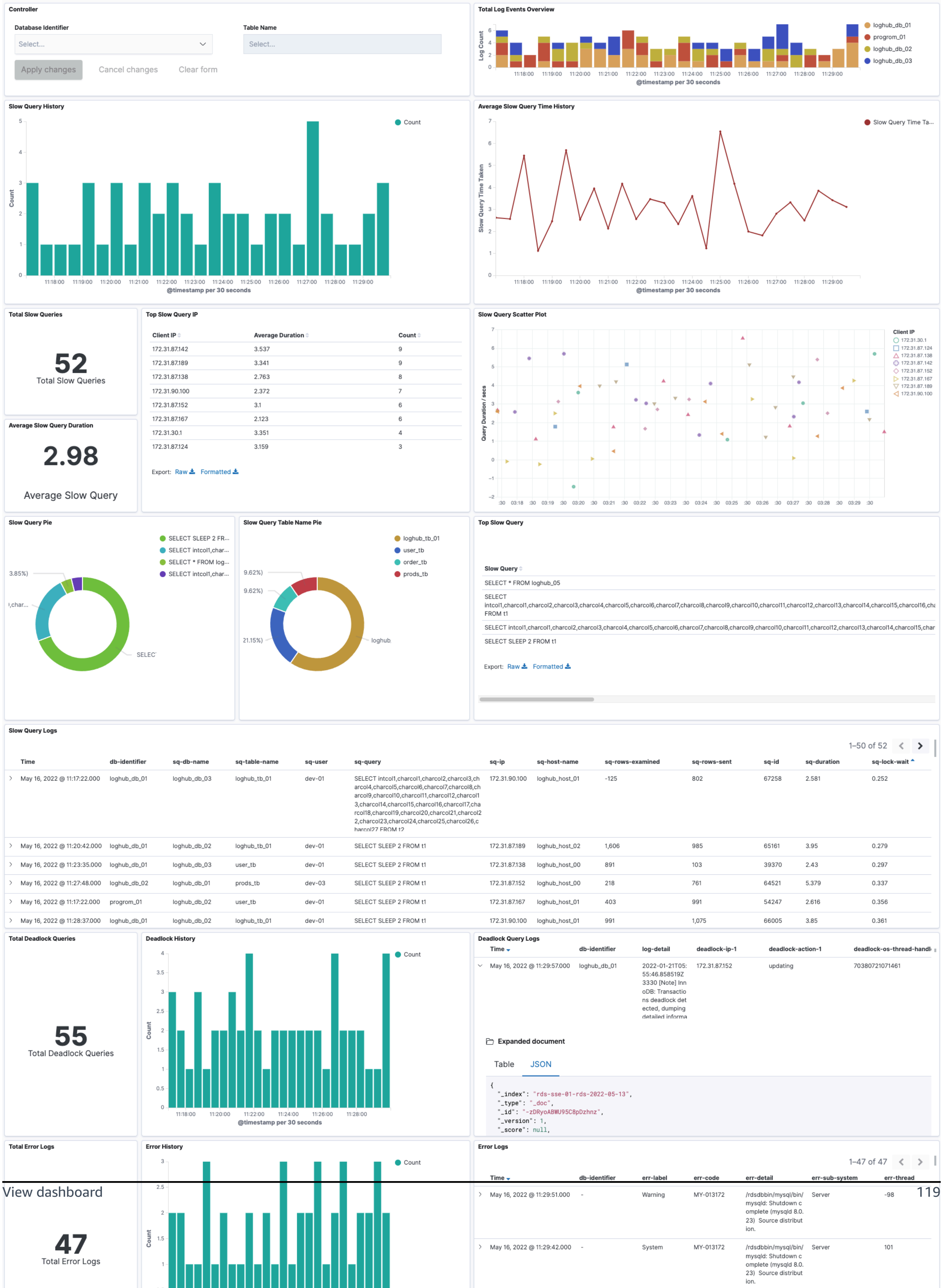
Visualization Name	Source Field	Description
Error Logs	• db-identifier • err-label • err-code • err-detail • err-sub-system • err-thread	This visualization displays the error logs generated by the AWS RDS instance. It provides valuable insights into any errors, warnings, or issues encountered within the database system, helping to identify and troublesh oot problems effectively. Monitoring error logs is essential for maintaining the health and reliability of the database.
Audit History	• log event	This visualization presents the audit history of the Amazon RDS instance. It tracks the various log events and activities related to database access, modifications, and security-related events. Monitoring the audit logs is crucial for ensuring complianc e, detecting unauthorized access, and keeping track of changes made to the database.

Visualization Name	Source Field	Description
Audit Logs	• db-identifier • audit-operation • audit-ip • audit-query • audit-retcode • audit-connection-id • audit-host-name • audit-query-id • audit-user	This visualization provides an overview of the audit logs generated by the Amazon RDS instance. It shows the operations performed on the database, including queries executed, connection details, IP addresses, and associated users. Monitoring audit logs enhances the security and governance of the database, helping to detect suspiciou s activities and track user actions.

Sample Dashboard

You can access the built-in dashboard in Amazon OpenSearch Service to view log data. For more information, see Access Dashboard.

You can click the below image to view the high-resolution sample dashboard.


Controller
Database Identifier
Table Name
Apply changes
Cancel changes
Clear form
Total Log Events Overview
Slow Query History
Average Slow Query Time History
Total Slow Queries
52
Total Slow Queries
Top Slow Query IP
Slow Query Scatter Plot
Average Slow Query Duration
2.98
Average Slow Query
Slow Query Pie
Slow Query Table Name Pie
Top Slow Query
Slow Query Logs
Total Deadlock Queries
55
Total Deadlock Queries
Deadlock History
Deadlock Query Logs
Total Error Logs
47
Total Error Logs
Error History
Error Logs

Amazon CloudFront Logs

CloudFront standard logs provide detailed records about every request made to a distribution.

You can create a log ingestion into Amazon OpenSearch Service or Light Engine either by using the Centralized Logging with OpenSearch console or by deploying a standalone CloudFormation stack.

> **Important**
>
> - The CloudFront logging bucket must be in the same Region as the Centralized Logging with OpenSearch solution.
> - The Amazon OpenSearch Service index is rotated on a daily basis by default, and you can adjust the index in the Additional Settings.

Create log ingestion (Amazon OpenSearch Service for log analytics)

Using the Centralized Logging with OpenSearch Console

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the navigation pane, under **Log Analytics Pipelines**, choose **Service Log**.
3. Choose the **Create a log ingestion** button.
4. In the **AWS Services** section, choose **Amazon CloudFront**.
5. Choose **Amazon OpenSearch Service**, and choose **Next**.
6. Under **Specify settings**, choose **Automatic** or **Manual** for **CloudFront logs enabling**. The automatic mode will detect the CloudFront log location automatically.
   - For **Automatic mode**, choose the CloudFront distribution and Log Type from the dropdown lists.
   - For Standard Log, the solution will automatically detect the log location if logging is enabled.
   - For Real-time log, the solution will prompt you for confirmation to create or replace CloudFront real-time log configuration.
   - For **Manual mode**, enter the **CloudFront Distribution ID** and **CloudFront Standard Log location**. (Note that CloudFront real-time log is not supported in Manual mode)
7. (Optional) If you are ingesting CloudFront logs from another account, select a linked account from the **Account** dropdown list first.
8. Choose **Next**.

9. In the **Specify OpenSearch domain** section, select an imported domain for **Amazon OpenSearch Service domain**.
10. Choose **Yes** for **Sample dashboard** if you want to ingest an associated templated Amazon OpenSearch Service dashboard.
11. You can change the **Index Prefix** of the target Amazon OpenSearch Service index if needed. The default prefix is the CloudFront distribution ID.
12. In the **Log Lifecycle** section, input the number of days to manage the Amazon OpenSearch Service index lifecycle. The Centralized Logging with OpenSearch will create the associated Index State Management (ISM) policy automatically for this pipeline.
13. In the **Log processor settings** section, choose **Log processor type**, configure the Lambda concurrency if needed, and then choose **Next**.
14. Add tags if needed.
15. Choose **Create**.

Using the CloudFormation Stack

This automated AWS CloudFormation template deploys the *Centralized Logging with OpenSearch - CloudFront Standard Log Ingestion* template in the AWS Cloud.

	Launch in AWS Management Console	Download Template
AWS Regions	Launch Stack	Template
AWS China Regions	Launch Stack	Template

1. Log in to the AWS Management Console and select above button to launch the AWS CloudFormation template. You can also download the template as a starting point for your own implementation.
2. To launch the stack in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.

5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following parameters.

Parameter	Default	Description
Log Bucket Name	*Requires input*	The S3 bucket name which stores the logs.
Log Bucket Prefix	*Requires input*	The S3 bucket path prefix which stores the logs.
Log Source Account ID	<Optional input>	The AWS Account ID of the S3 bucket. Required for cross-account log ingestion (Please add a member account first). By default, the Account ID you logged in at **Step 1** will be used.
Log Source Region	<Optional>	The AWS Region of the S3 bucket. By default, the Region you selected at **Step 2** will be used.
Log Source Account Assume Role	<Optional>	The IAM Role ARN used for cross-account log ingestion . Required for cross-account log ingestion (Please add a member account first).
Engine Type	OpenSearch	The engine type of the OpenSearch. Select OpenSearch.
OpenSearch Domain Name	*Requires input*	The domain name of the Amazon OpenSearch Service cluster.

Parameter	Default	Description
OpenSearch Endpoint	*Requires input*	The OpenSearch endpoint URL. For example, vpc-your_opensearch_domain_name-xcvgw6uu2o6zafsiefxubwuohe.us-east-1.es.amazonaws.com
Index Prefix	*Requires input*	The common prefix of OpenSearch index for the log. The index name will be <Index Prefix>-<Log Type>-<Other Suffix>.
Create Sample Dashboard	Yes	Whether to create a sample OpenSearch dashboard.
VPC ID	*Requires input*	Select a VPC which has access to the OpenSearch domain. The log processor Lambda function will reside in the selected VPC.
Subnet IDs	*Requires input*	Select at least two subnets which have access to the OpenSearch domain. The log processor Lambda function will reside in the subnets. Make sure the subnets have access to the Amazon S3 service.

Parameter	Default	Description
Security Group ID	*Requires input*	Select a Security Group which will be associated with the log processor Lambda function. Make sure the Security Group has access to the OpenSearch domain.
S3 Backup Bucket	*Requires input*	The S3 backup bucket name to store the failed ingestion logs.
KMS-CMK ARN	<Optional>	The KMS-CMK ARN for encryption. Leave it blank to create a new KMS CMK.
Number Of Shards	5	Number of shards to distribute the index evenly across all data nodes. Keep the size of each shard between 10-50 GB.
Number of Replicas	1	Number of replicas for OpenSearch Index. Each replica is a full copy of an index.
Age to Warm Storage	<Optional>	The age required to move the index into warm storage (e.g. 7d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). This is only effective when warm storage is enabled in OpenSearch.

Parameter	Default	Description
Age to Cold Storage	<Optional>	The age required to move the index into cold storage (e.g. 30d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). This is only effective when cold storage is enabled in OpenSearch.
Age to Retain	<Optional>	The age to retain the index (e.g. 180d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). If value is "", the index will not be deleted.
Rollover Index Size	<Optional>	The minimum size of the shard storage required to roll over the index (e.g. 30GB).
Index Suffix	yyyy-MM-dd	The common suffix format of OpenSearch index for the log(Example: yyyy-MM-dd, yyyy-MM-dd-HH). The index name will be <Index Prefix>-<Log Type>-<Index Suffix>-000001.
Compression type	best_compression	The compression type to use to compress stored data. Available values are best_compression and default.

Parameter	Default	Description
Refresh Interval	1s	How often the index should refresh, which publishes its most recent changes and makes them available for searching. Can be set to -1 to disable refreshing. Default is 1s.
Plugins	<Optional>	List of plugins delimited by comma. Leave it blank if there are no available plugins to use. Valid inputs are user_agent, geo_ip.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template creates AWS Identity and Access Management (IAM) resources.
9. Choose **Create** stack to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately 10 minutes.

View dashboard

The dashboard includes the following visualizations.

Visualization Name	Source Field	Description
Total Requests	• log event	Displays the total number of viewer requests received by the Amazon CloudFron t, for all HTTP methods and for both HTTP and HTTPS requests.

Visualization Name	Source Field	Description
Edge Locations	• x-edge-location	Shows a pie chart represent ing the proportion of the locations of CloudFront edge servers.
Request History	• log event	Presents a bar chart that displays the distribution of events over time.
Unique Visitors	• c-ip	Displays unique visitors identified by client IP address.
Cache Hit Rate	• sc-bytes	Shows the proportion of your viewer requests that are served directly from the CloudFront cache instead of going to your origin servers for content.

Visualization Name	Source Field	Description
Result Type	• x-edge-response-result-type	Shows the percentage of hits, misses, and errors to the total viewer requests for the selected CloudFront distribution: • Hit – A viewer request for which the object is served from a CloudFront edge cache. In access logs, these are requests for which the value of x-edge-response-result-type is Hit • Miss – A viewer request for which the object isn't currently in an edge cache, so CloudFront must get the object from your origin. In access logs, these are requests for which the value of x-edge-response-result-type is Miss. • Error – A viewer request that resulted in an error, so CloudFront didn't serve the object. In access logs, these are requests for which the value of x-edge-response-result-type is Error, LimitExceeded, or CapacityExceeded. The chart does not include refresh hits—requests for

Visualization Name	Source Field	Description
		objects that are in the edge cache but that have expired. In access logs, refresh hits are requests for which the value of x-edge-response-result-type is RefreshHit.
Top Miss URI	• cs-uri-stem • cs-method	Shows top 10 of the requested objects that are not in the cache.
Bandwidth	• cs-bytes • sc-bytes	Provides insights into data transfer activities from the locations of CloudFront edge.
Bandwidth History	• cs-bytes • sc-bytes	Shows the historical trend of the data transfer activitie s from the locations of CloudFront edge.
Top Client IPs	• c-ip	Provides the top 10 IP address accessing your Amazon CloudFront.
Status Code Count	• sc-status	Displays the count of requests made to the Amazon CloudFront, grouped by HTTP status codes(e.g., 200, 404, 403, etc.).
Status History	• @timestamp • sc-status	Shows the historical trend of HTTP status codes returned by the Amazon CloudFront over a specific period of time.

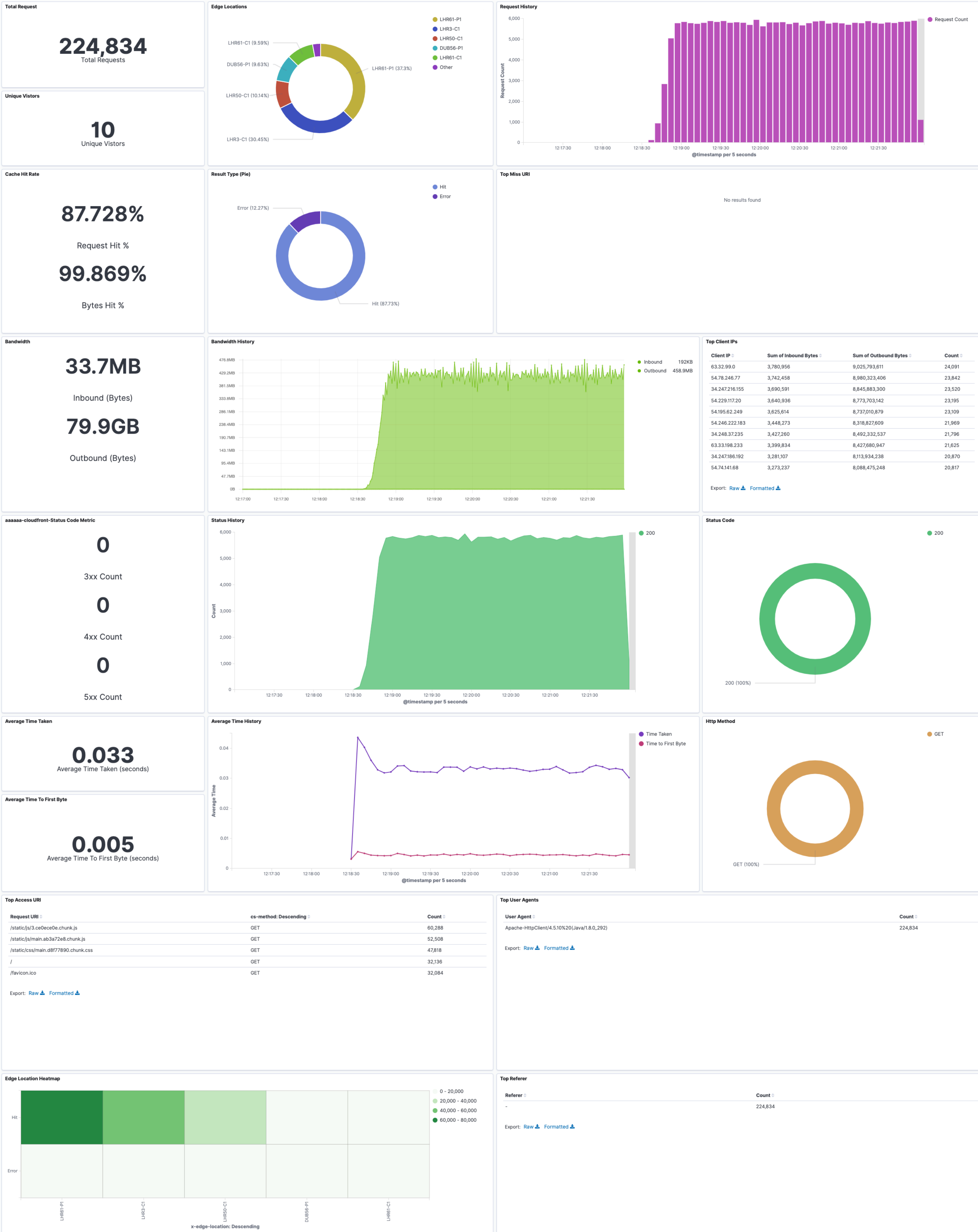
Visualization Name	Source Field	Description
Status Code	• sc-status	Identifies the users or IAM roles responsible for changes to EC2 resources, assisting in accountability and tracking of modifications.
Average Time Taken	• time-taken	This visualization calculate s and presents the average time taken for various operations in the Amazon CloudFront (e.g., average time for GET, PUT requests, etc.).
Average Time History	• time-taken • time-to-first-byte • @timestamp	Shows the historical trend of the average time taken for various operations in the Amazon CloudFront.
Http Method	• cs-method	Displays the count of requests made to the Amazon CloudFront using a pie chart, grouped by http request method names (e.g., POST, GET, HEAD, etc.).
Average Time To First Byte	• time-to-first-byte	Provides the average time taken in seconds by the origin server to respond back with the first byte of the response.
Top Request URIs	• cs-uri-stem • cs-method	Provides the top 10 request URIs accessing your CloudFront.

Visualization Name	Source Field	Description
Top User Agents	• cs-user-agent	Provides the top 10 user agents accessing your CloudFront.
Edge Location Heatmap	• x-edge-location • x-edge-result-type	Shows a heatmap represent ing the result type of each edge locations.
Top Referers	• cs-referer	Top 10 referers with the Amazon CloudFront access.
Top Countries or Regions	• c_country	Top 10 countries with the Amazon CloudFront access.

Sample dashboard

You can access the built-in dashboard in Amazon OpenSearch Service to view log data. For more information, see Access Dashboard.

You can click the below image to view the high-resolution sample dashboard.


Total Request
224,834
Total Requests
Unique Vistors
10
Unique Vistors
Cache Hit Rate
87.728%
Request Hit %
99.869%
Bytes Hit %
Edge Locations
LHR61-P1
LHR3-C1
LHR50-C1
DUB56-P1
LHR61-C1
Other
LHR61-C1 (9.59%)
DUB56-P1 (9.63%)
LHR50-C1 (10.14%)
LHR3-C1 (30.45%)
LHR61-P1 (37.3%)
Request History
Request Count
@timestamp per 5 seconds
Result Type (Pie)
Hit
Error
Error (12.27%)
Hit (87.73%)
Top Miss URI
No results found
Bandwidth
33.7MB
Inbound (Bytes)
79.9GB
Outbound (Bytes)
Bandwidth History
Inbound 192KB
Outbound 458.9MB
Top Client IPs
Client IP | Sum of Inbound Bytes | Sum of Outbound Bytes | Count
63.32.99.0 | 3,780,956 | 9,025,793,611 | 24,091
54.78.246.77 | 3,742,458 | 8,980,323,406 | 23,842
34.247.216.155 | 3,690,591 | 8,845,883,300 | 23,520
54.229.117.20 | 3,640,936 | 8,773,703,142 | 23,195
54.195.62.249 | 3,625,614 | 8,737,010,879 | 23,109
54.246.222.183 | 3,448,273 | 8,318,827,609 | 21,969
34.248.37.235 | 3,427,260 | 8,492,332,537 | 21,796
63.33.198.233 | 3,399,834 | 8,427,680,947 | 21,625
34.247.186.192 | 3,281,107 | 8,113,934,238 | 20,870
54.74.141.68 | 3,273,237 | 8,088,475,248 | 20,817
Export: Raw Formatted
aaaaaa-cloudfront-Status Code Metric
0
3xx Count
0
4xx Count
0
5xx Count
Status History
Count
200
@timestamp per 5 seconds
Status Code
200
200 (100%)
Average Time Taken
0.033
Average Time Taken (seconds)
Average Time To First Byte
0.005
Average Time To First Byte (seconds)
Average Time History
Time Taken
Time to First Byte
Average Time
@timestamp per 5 seconds
Http Method
GET
GET (100%)
Top Access URI
Request URI | cs-method: Descending | Count
/static/js/3.ce0ece0e.chunk.js | GET | 60,288
/static/js/main.ab3a72e8.chunk.js | GET | 52,508
/static/css/main.d8f77890.chunk.css | GET | 47,818
/ | GET | 32,136
/favicon.ico | GET | 32,084
Export: Raw Formatted
Top User Agents
User Agent | Count
Apache-HttpClient/4.5.10%20(Java/1.8.0_292) | 224,834
Export: Raw Formatted
Edge Location Heatmap
0 - 20,000
20,000 - 40,000
40,000 - 60,000
60,000 - 80,000
Hit
Error
LHR61-P1
LHR3-C1
LHR50-C1
DUB56-P1
LHR61-C1
x-edge-location: Descending
Top Referer
Referer | Count
- | 224,834
Export: Raw Formatted

Create log ingestion (Light Engine for log analytics)

Using the Centralized Logging with OpenSearch Console

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the navigation pane, under **Log Analytics Pipelines**, choose **Service Log**.
3. Choose the **Create a log ingestion** button.
4. In the **AWS Services** section, choose **Amazon CloudFront**.
5. Choose **Light Engine**, and choose **Next**.
6. Under **Specify settings**, choose **Automatic** or **Manual** for **CloudFront logs enabling**. The automatic mode will detect the CloudFront log location automatically.
   - For **Automatic mode**, choose the CloudFront distribution and Log Type from the dropdown lists.
   - For Standard Log, the solution will automatically detect the log location if logging is enabled.
   - For **Manual mode**, enter the **CloudFront Distribution ID** and **CloudFront Standard Log location**. (Note that CloudFront real-time log is not supported in Manual mode)
   - (Optional) If you are ingesting CloudFront logs from another account, select a linked account from the **Account** dropdown list first.
7. Choose **Next**.
8. Choose **Log Processing Enriched fields** if needed. The available plugins are **location** and **OS/User Agent**. Enabling rich fields increases data processing latency and processing cost. By default, it is not selected.
9. In the **Specify Light Engine Configuration** section, if you want to ingest an associated templated Grafana dashboard, select **Yes** for the sample dashboard.
10. Choose an existing Grafana, or import a new one by making configurations in Grafana.
11. Select an Amazon S3 bucket to store partitioned logs and give a name to the log table. The solution provides a predefined table name, but you can modify it according to your needs.
12. Modify the log processing frequency if needed, which is set to **5** minutes by default with a minimum processing frequency of **1** minute.
13. In the **Log Lifecycle** section, if needed, enter the log merge time and log archive time to modify the default values provided by the solution.
14. Choose **Next**.
15. Add tags if needed.
16. Choose **Create**.

Using the CloudFormation Stack

This automated AWS CloudFormation template deploys the *Centralized Logging with OpenSearch - CloudFront Standard Log Ingestion* template in the AWS Cloud.

	Launch in AWS Management Console	Download Template
AWS Regions	Launch Stack	Template
AWS China Regions	Launch Stack	Template

1. Log in to the AWS Management Console and select above button to launch the AWS CloudFormation template. You can also download the template as a starting point for your own implementation.
2. To launch the stack in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following parameters.
   - Parameters for **Pipeline settings**

Parameter	Default	Description
Pipeline Id	`Requires input`	The unique identifier for the pipeline, which is essential if you need to create multiple ALB pipelines and write different ALB logs into separate tables. To ensure uniqueness, you can

Parameter	Default	Description
		generate a unique pipeline identifier using uuidgenerator.
Staging Bucket Prefix	AWSLogs/CloudFrontLogs	The storage directory for logs in the temporary storage area should ensure uniqueness and non-overlapping of the prefix for different pipelines.

- Parameters for **Destination settings**

Parameter	Default	Description
Centralized Bucket Name	*`Requires input`*	The name for the centralized S3 bucket. For example, `centralized-logging-bucket` .
Centralized Bucket Prefix	datalake	The centralized bucket prefix. By default, the database location is s3://{Centralized Bucket Name}/{Centralized Bucket Prefix}/amazon_cl_centralized.
Centralized Table Name	CloudFront	Table name for writing data to the centralized database. You can modify it if needed.

Parameter	Default	Description
Enrichment Plugins	<Optional input>	The available plugins to choose from are **location** and **OS/User Agent** . Enabling rich fields will increase data processin g latency and processing costs. It is not selected by default.

- Parameters for **Scheduler settings**

Parameter	Default	Description
LogProcessor Schedule Expression	rate (5 minutes)	Task scheduling expression for performing log processin g, with a default value of executing the LogProcessor every 5 minutes. For more information, see Schedule types.
LogMerger Schedule Expression	cron(0 1 * ?)	Task scheduling expression for performing log merging, with a default value of executing the LogMerger at 1 AM every day. For more information, see Schedule types.

Parameter	Default	Description
LogArchive Schedule Expression	cron(0 2 * ?)	Task scheduling expression for performing log archiving , with a default value of executing the LogArchive at 2 AM every day. For more information, see Schedule types.
Age to Merge	7	Small file retention days, with a default value of 7, indicating that logs older than 7 days will be merged into small files. It can be adjusted as needed.
Age to Archive	30	Log retention days, with a default value of 30, indicating that data older than 30 days will be archived and deleted. It can be adjusted as needed.

- Parameters for **Notification settings**

Parameter	Default	Description
Notification Service	SNS	Notification method for alerts. If your main stack is in AWS China Regions, you can only choose the SNS method. If your main stack is in AWS Regions, you can choose either the SNS or SES method.
Recipients	*Requires input*	If the Notification Service is SNS, enter the SNS Topic ARN to ensure that you have the required permissions. If the Notification Service is SES, enter the email addresses separated by commas to ensure that the email addresses are already Verified Identities in SES. The adminEmail provided during the creation of the main stack will receive a verification email by default.

- Parameters for **Dashboard settings**

Parameter	Default	Description
Import Dashboards	FALSE	Whether to import the Dashboard into Grafana. If it is set to `true`, you must provide the Grafana URL and Grafana Service Account Token.
Grafana URL	*`Requires input`*	Grafana access URL. For example, `https://alb-72277319.us-west-2.elb.amazonaws.com` .
Grafana Service Account Token	*`Requires input`*	Service Account Token created in Grafana.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template creates AWS Identity and Access Management (IAM) resources.
9. Choose **Create** stack to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately 10 minutes.

View dashboard

Visualization Name	Source Field	Description
Filters	Filters	The following data can be filtered by query filter conditions.

Visualization Name	Source Field	Description
Total Requests	log event	Displays the total number of viewer requests received by the Amazon CloudFron t, for all HTTP methods and for both HTTP and HTTPS requests.
Unique Visitors	c-ip	Displays unique visitors identified by client IP address.
Requests History	log event	Presents a bar chart that displays the distribution of events over time.
Request By Edge Location	x-edge-location	Shows a pie chart represent ing the proportion of the locations of CloudFront edge servers.
HTTP Status Code	sc-status	Displays the count of requests made to the Amazon CloudFront, grouped by HTTP status codes (for example, 200, 404, 403).
Status Code History	sc-status	Shows the historical trend of HTTP status codes returned by the Amazon CloudFront over a specific period of time.
Status Code Pie	sc-status	Represents the distribution of requests based on different HTTP status codes using a pie chart.

Visualization Name	Source Field	Description
Average Processing Time	time-taken time-to-first-byte	This visualization calculate s and presents the average time taken for various operations in the Amazon CloudFront (for example, average time for GET, and PUT requests).
Avg. Processing Time History	time-taken time-to-first-byte	Shows the historical trend of the average time taken for various operations in the Amazon CloudFront.
Avg. Processing Time History	time-taken time-to-first-byte	Shows the historical trend of the average time taken for various operations in the Amazon CloudFront.
HTTP Method	cs-method	Displays the count of requests made to the Amazon CloudFront using a pie chart, grouped by HTTP request method names (for example, POST, GET, and HEAD).
Total Bytes	cs-bytes sc-bytes	Provides insights into data transfer activities, including the total bytes transferred.
Response Bytes History	cs-bytes sc-bytes	Displays the historical trend of the received bytes, send bytes.

Visualization Name	Source Field	Description
Edge Response Type	x-edge-response-result-type	Shows the percentage of hits, misses, and errors to the total viewer requests for the selected CloudFront distribut ion: Hit – A viewer request for which the object is served from a CloudFront edge cache. In access logs, these are requests for which the value of x-edge-response-re sult-type is Hit. Miss – A viewer request for which the object isn't currently in an edge cache, so CloudFront must get the object from your origin. In access logs, these are requests for which the value of x-edge-response-result-type is Miss. Error – A viewer request that resulted in an error, so CloudFront didn't serve the object. In access logs, these are requests for which the value of x-edge-response-re sult-type is Error, LimitExce eded, or CapacityExceeded. The chart does not include refresh hits, that is, requests for objects that are in the edge cache but that have

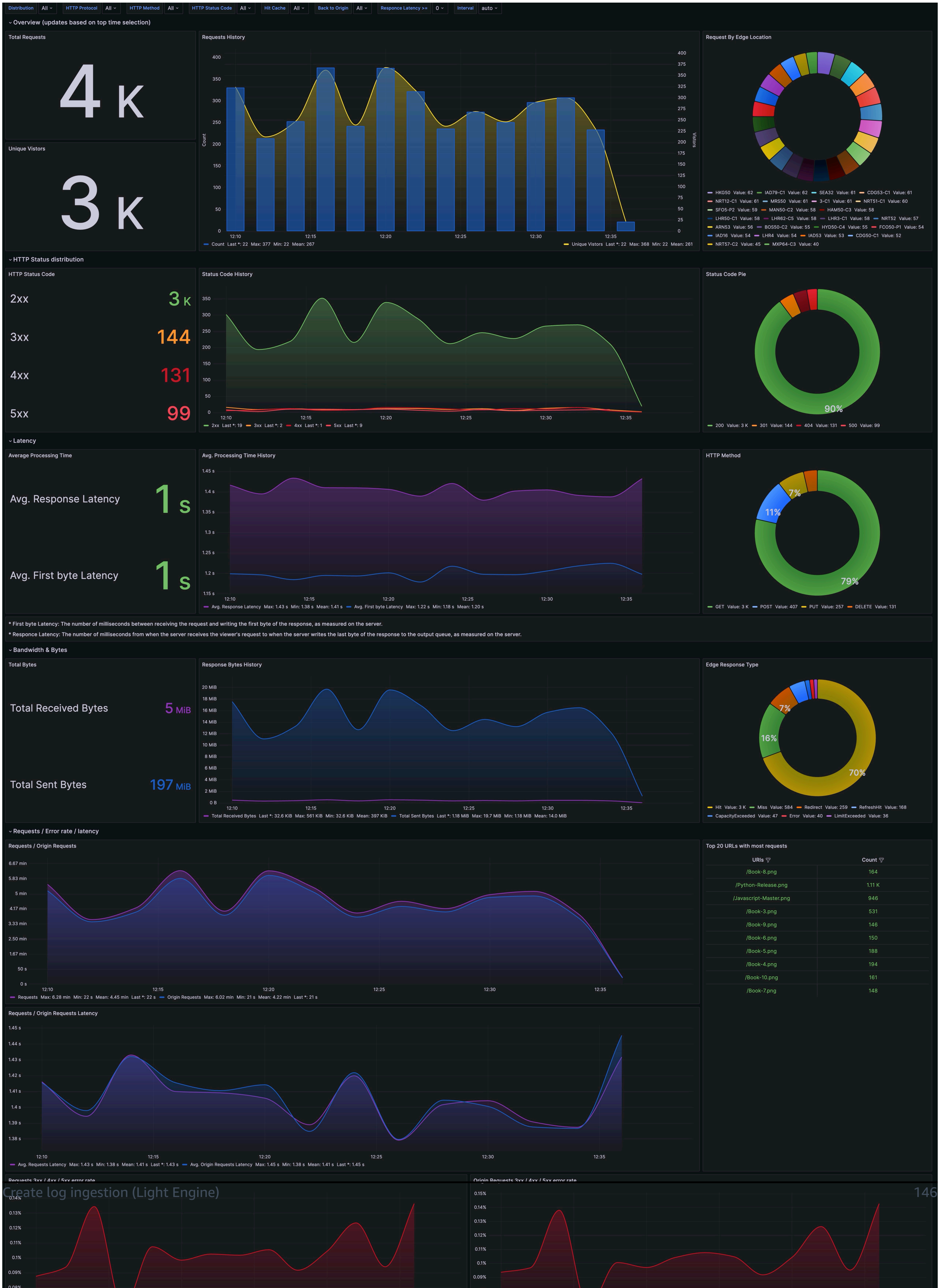
Visualization Name	Source Field	Description
		expired. In access logs, refresh hits are requests for which the value of x-edge-response-re sult-type is RefreshHit.
Requests / Origin Requests	log event	Displays the number of requests made to CloudFron t and the number of requests back to the origin.
Requests / Origin Requests Latency	log event time-taken	Displays the request latency from the client to CloudFront and the request latency back to the origin.
Top 20 URLs with most requests	log event	Top 20 URLs based on the number of requests.
Requests 3xx / 4xx / 5xx error rate	log event sc-status	Displays the ratio of 3xx/4xx/5xx status codes from the client to CloudFront.
Origin Requests 3xx / 4xx / 5xx error rate	log event sc-status x-edge-de tailed-result-type	Display the proportion of 3xx/4xx/5xx status codes returned to the origin.
Requests 3xx / 4xx / 5xx error latency	log event sc-status time-take n	Displays the latency from the client to CloudFront for 3xx/4xx/5xx status codes.
Origin Requests 3xx / 4xx / 5xx error latency	log event sc-status x-edge-de tailed-result-type time-taken	Displays the delay in returning to the source 3xx/4xx/5xx status code.
Response Latency (>= 1sec) rate	log event time-taken	Display the proportion of delay above 1s.

Visualization Name	Source Field	Description
Bandwidth	sc-bytes	Displays the bandwidth from the client to CloudFront and the bandwidth back to the origin.
Data transfer	sc-bytes	Display the response traffic.
Top 20 URLs with most traffic	cs-uri-stem sc-bytes	Top 20 URLs calculated by traffic.
Cache hit rate (calculated using requests)	log event x-edge-result-type	Displays the cache hit ratio calculated by the number of requests.
Cache hit rate (calculated using bandwidth)	log event sc-bytes x-edge-re sult-type	Displays the cache hit ratio calculated by bandwidth.
Cache Result	log event x-edge-result-type	Displays the number of requests of various x-edge-result-types, such as the number of requests that hit the cache and the number of requests that missed the cache.
Cache Result Latency	log event sc-bytes x-edge-re sult-type	Displays the request latency of various x-edge-result-type s, such as the request latency that hits the cache and the request latency that misses the cache.
Requests by OS	ua_os	Displays the count of requests made to the ALB, grouped by user agent OS.

Visualization Name	Source Field	Description
Requests by Device	ua_device	Displays the count of requests made to the ALB, grouped by user agent device.
Requests by Browser	ua_browser	Displays the count of requests made to the ALB, grouped by user agent browser.
Requests by Category	ua_category	Displays the count of category made to the ALB, grouped by user agent category (for example, PC, Mobile, Tablet).
Requests by Countries or Regions	geo_iso_code	Displays the count of requests made to the ALB (grouped by the corresponding country or region resolved by the client IP).
Top Countries or Regions	geo_country	Top 10 countries with the ALB Access.
Top Cities	geo_city	Top 10 cities with ALB Access.

Sample dashboard

Below shows the sample dashboard.


Overview (updates based on top time selection)
Total Requests
4 K
Requests History
Request By Edge Location
Unique Vistors
3 K
HTTP Status distribution
HTTP Status Code
2xx 3 K
3xx 144
4xx 131
5xx 99
Status Code History
Status Code Pie
Latency
Average Processing Time
Avg. Response Latency 1 s
Avg. First byte Latency 1 s
Avg. Processing Time History
HTTP Method
* First byte Latency: The number of milliseconds between receiving the request and writing the first byte of the response, as measured on the server.
* Responce Latency: The number of milliseconds from when the server receives the viewer's request to when the server writes the last byte of the response to the output queue, as measured on the server.
Bandwidth & Bytes
Total Bytes
Total Received Bytes 5 MiB
Total Sent Bytes 197 MiB
Response Bytes History
Edge Response Type
Requests / Error rate / latency
Requests / Origin Requests
Top 20 URLs with most requests
Requests / Origin Requests Latency

AWS Lambda Logs

AWS Lambda automatically monitors Lambda functions on your behalf and sends function metrics to Amazon CloudWatch.

Create log ingestion

You can create a log ingestion into Amazon OpenSearch Service either by using the Centralized Logging with OpenSearch console or by deploying a standalone CloudFormation stack.

Important

- The Lambda Region must be the same as the Centralized Logging with OpenSearch solution Region.
- The Amazon OpenSearch Service index is rotated on a daily basis by default, and you can adjust the index in the Additional Settings.

Using the Centralized Logging with OpenSearch Console

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the navigation pane, under **Log Analytics Pipelines**, choose **Service Log**.
3. Choose the **Create a log ingestion** button.
4. In the **AWS Services** section, choose **AWS Lambda**.
5. Choose **Next**.
6. Under **Specify settings**, choose the Lambda function from the dropdown list. (Optional) If you are ingesting logs from another account, select a linked account from the **Account** dropdown first.
7. Choose **Next**.
8. In the **Specify OpenSearch domain** section, select an imported domain for **Amazon OpenSearch Service domain**.
9. Choose **Yes** for **Sample dashboard** if you want to ingest an associated templated Amazon OpenSearch Service dashboard.
10. You can change the **Index Prefix** of the target Amazon OpenSearch Service index if needed. The default prefix is the Lambda function name.

11. In the **Log Lifecycle** section, input the number of days to manage the Amazon OpenSearch Service index lifecycle. The Centralized Logging with OpenSearch will create the associated [Index State Management (ISM)](#) policy automatically for this pipeline.
12. Choose **Next**.
13. Add tags if needed.
14. Choose **Create**.

Using the CloudFormation Stack

This automated AWS CloudFormation template deploys the *Centralized Logging with OpenSearch - Lambda Log Ingestion* solution in the AWS Cloud.

	Launch in AWS Management Console	**Download Template**
AWS Regions	Launch Stack	Template
AWS China Regions	Launch Stack	Template

1. Log in to the AWS Management Console and select the button to launch the AWS CloudFormation template. You can also download the template as a starting point for your own implementation.
2. To launch the Centralized Logging with OpenSearch in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following parameters.

Parameter	Default	Description
Log Bucket Name	*Requires input*	The S3 bucket name to export the logs.

Parameter	Default	Description
Log Bucket Prefix	*Requires input*	The S3 bucket path prefix which stores the the logs.
Log Source Account ID	<Optional>	The AWS Account ID of the CloudWatch log group. Required for cross-account log ingestion (Please add a member account first). By default, the Account ID you logged in at **Step 1** will be used.
Log Source Region	<Optional input>	The AWS Region of the CloudWatch log group. By default, the Region you selected at **Step 2** will be used.
Log Source Account Assume Role	<Optional input>	The IAM Role ARN used for cross-account log ingestion . Required for cross-account log ingestion (Please add a member account first).
Log Group Names	*Requires input*	The names of the CloudWatc h log group for the logs.
Engine Type	OpenSearch	The engine type of the OpenSearch. Select OpenSearch.
OpenSearch Domain Name	*Requires input*	The domain name of the Amazon OpenSearch Service cluster.

Parameter	Default	Description
OpenSearch Endpoint	*Requires input*	The OpenSearch endpoint URL. For example, vpc-your_opensearch_domain_name-xcvgw6uu2o6zafsiefxubwuohe.us-east-1.es.amazonaws.com
Index Prefix	*Requires input*	The common prefix of OpenSearch index for the log. The index name will be <Index Prefix>-<Log Type>-<Other Suffix>.
Create Sample Dashboard	Yes	Whether to create a sample OpenSearch dashboard.
VPC ID	*Requires input*	Select a VPC which has access to the OpenSearch domain. The log processor Lambda function will reside in the selected VPC.
Subnet IDs	*Requires input*	Select at least two subnets which has access to the OpenSearch domain. The log processor Lambda function will reside in the subnets. Please make sure the subnets has access to the Amazon S3 service.

Parameter	Default	Description
Security Group ID	*Requires input*	Select a Security Group which will be associated to the log processor Lambda function. Please make sure the Security Group has access to the OpenSearch domain.
S3 Backup Bucket	*Requires input*	The S3 backup bucket name to store the failed ingestion logs.
KMS-CMK ARN	<Optional input>	The KMS-CMK ARN for SQS encryption. Leave it blank to create a new KMS CMK.
Number Of Shards	5	Number of shards to distribute the index evenly across all data nodes. Keep the size of each shard between 10-50 GiB.
Number of Replicas	1	Number of replicas for OpenSearch Index. Each replica is a full copy of an index.
Age to Warm Storage	<Optional>	The age required to move the index into warm storage (e.g. 7d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). This is only effective when warm storage is enabled in OpenSearch.

Parameter	Default	Description
Age to Cold Storage	<Optional>	The age required to move the index into cold storage (e.g. 30d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). This is only effective when cold storage is enabled in OpenSearch.
Age to Retain	<Optional>	The age to retain the index (e.g. 180d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). If value is "", the index will not be deleted.
Rollover Index Size	<Optional>	The minimum size of the shard storage required to roll over the index (e.g. 30GB).
Index Suffix	yyyy-MM-dd	The common suffix format of OpenSearch index for the log(Example: yyyy-MM-dd, yyyy-MM-dd-HH). The index name will be <Index Prefix>-<Log Type>-<Index Suffix>-000001.
Compression type	best_compression	The compression type to use to compress stored data. Available values are best_compression and default.

Parameter	Default	Description
Refresh Interval	1s	How often the index should refresh, which publishes its most recent changes and makes them available for searching. Can be set to -1 to disable refreshing. Default is 1s.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template creates AWS Identity and Access Management (IAM) resources.
9. Choose **Create** stack to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately 15 minutes.

View dashboard

The dashboard includes the following visualizations.

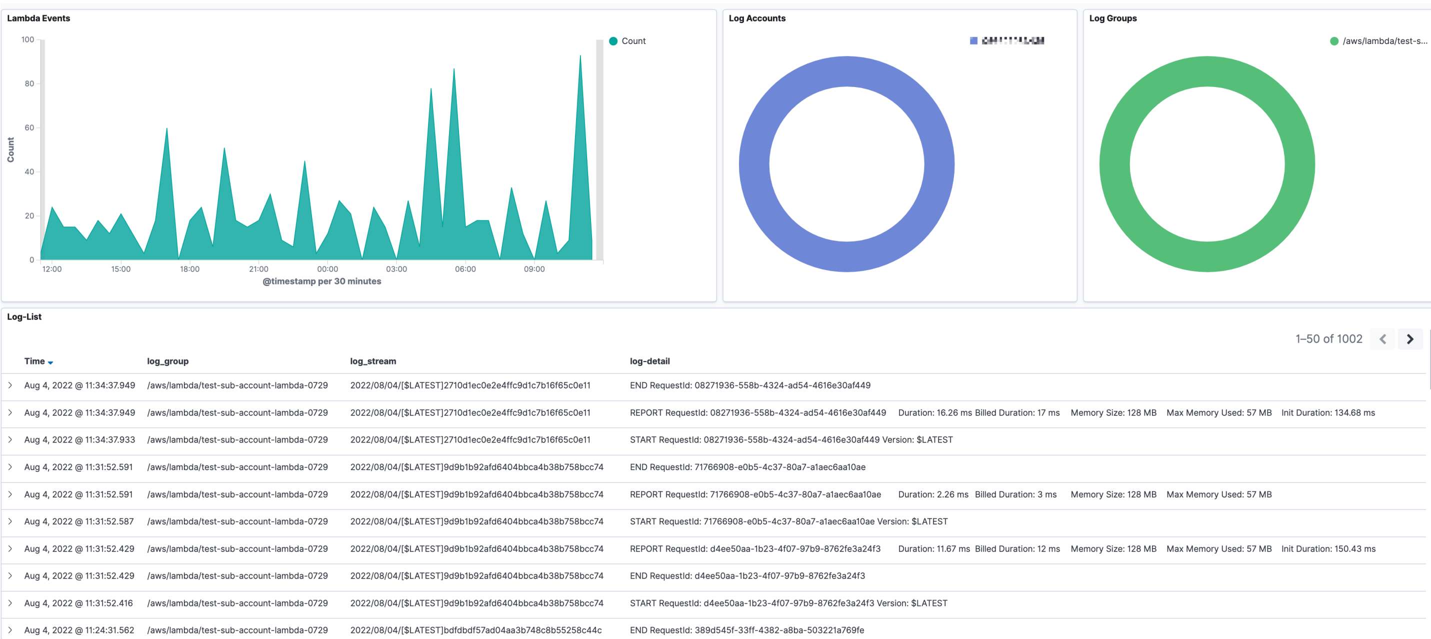
Visualization Name	Source Field	Description
Lambda Events	• log event	Presents a chart that displays the distribution of events over time.
Log Accounts	• owner	Shows a pie chart represent ing the proportion of log events from different AWS accounts (owners).
Log Groups	• log_group	Displays a pie chart depicting the distribution of log events

Visualization Name	Source Field	Description
		among various log groups in the Lambda environment.
Log-List	• time • log_group • log_stream • log_detail	Provides a detailed list of log events, including timestamps, log groups, log streams, and log details.

Sample Dashboard

You can access the built-in dashboard in Amazon OpenSearch Service to view log data. For more information, see Access Dashboard.

You can click the below image to view the high-resolution sample dashboard.



Elastic Load Balancing access logs

Elastic Load Balancing access logs provide access logs that capture detailed information about requests sent to your load balancer. ALB publishes a log file for each load balancer node every 5 minutes.

You can create a log ingestion into Amazon OpenSearch Service or Light Engine either by using the Centralized Logging with OpenSearch console or by deploying a standalone CloudFormation stack.

Important

- The ALB logging bucket must be in the same region as the Centralized Logging with OpenSearch solution.
- The Amazon OpenSearch Service index is rotated on a daily basis by default, and you can adjust the index in the Advanced Settings.

Create log ingestion (Amazon OpenSearch Service for log analytics)

Using the Centralized Logging with OpenSearch Console

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the navigation pane, under **Log Analytics Pipelines**, choose **Service Log**.
3. Choose the **Create a log ingestion** button.
4. In the **AWS Services** section, choose **Elastic Load Balancing**.
5. Choose **Amazon OpenSearch Service**, and choose **Next**.
6. Under **Specify settings**, choose **Automatic** or **Manual**.
   - For **Automatic** mode, choose an application load balancer in the dropdown list. (If the selected ALB access log is not enabled, click **Enable** to enable the ALB access log.)
   - For **Manual** mode, enter the **Application Load Balancer identifier** and **Log location**.
   - (Optional) If you are ingesting logs from another account, select a linked account from the **Account** dropdown first.
7. Choose **Next**.
8. In the **Specify OpenSearch domain** section, select an imported domain for **Amazon OpenSearch Service domain**.
9. Choose **Yes** for **Sample dashboard** if you want to ingest an associated templated Amazon OpenSearch Service dashboard.
10. You can change the **Index Prefix** of the target Amazon OpenSearch Service index if needed. The default prefix is the Load Balancer Name.
11. In the **Log Lifecycle** section, input the number of days to manage the Amazon OpenSearch Service index lifecycle. The Centralized Logging with OpenSearch will create the associated Index State Management (ISM) policy automatically for this pipeline.
12. In the **Select log processor** section, choose the log processor.

- When selecting Lambda as log processor, you can configure the Lambda concurrency if needed.
- (Optional) OSI as log processor is now supported in these Regions. When OSI is selected, enter the minimum and maximum number of OCU. For more information, see Scaling pipelines.

13. Choose **Next**.
14. Add tags if needed.
15. Choose **Create**.

Using the CloudFormation Stack

This automated AWS CloudFormation template deploys the *Centralized Logging with OpenSearch - ALB Log Ingestion* solution in the AWS Cloud.

	Launch in AWS Management Console	**Download Template**
AWS Regions	Launch Stack	Template
AWS China Regions	Launch Stack	Template

1. Log in to the AWS Management Console and select above button to launch the AWS CloudFormation template. You can also download the template as a starting point for your own implementation.
2. To launch the stack in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following parameters.

Parameter	Default	Description
Log Bucket Name	`Requires input`	The S3 bucket name which stores the logs.
Log Bucket Prefix	`Requires input`	The S3 bucket path prefix which stores the logs.
Log Source Account ID	<Optional input>	The AWS Account ID of the S3 bucket. Required for cross-account log ingestion (Please add a member account first). By default, the Account ID you logged in at **Step 1** will be used.
Log Source Region	<Optional>	The AWS Region of the S3 bucket. By default, the Region you selected at **Step 2** will be used.
Log Source Account Assume Role	<Optional>	The IAM Role ARN used for cross-account log ingestion . Required for cross-account log ingestion (Please add a member account first).
Engine Type	OpenSearch	The engine type of the OpenSearch. Select OpenSearch.
OpenSearch Domain Name	`Requires input`	The domain name of the Amazon OpenSearch Service cluster.

Parameter	Default	Description
OpenSearch Endpoint	`Requires input`	The OpenSearch endpoint URL. For example, vpc-your_opensearch_domain_name-xcvgw6uu2o6zafsiefxubwuohe.us-east-1.es.amazonaws.com
Index Prefix	`Requires input`	The common prefix of OpenSearch index for the log. The index name will be <Index Prefix>-<Log Type>-<Other Suffix>.
Create Sample Dashboard	Yes	Whether to create a sample OpenSearch dashboard.
VPC ID	`Requires input`	Select a VPC which has access to the OpenSearch domain. The log processor Lambda function will reside in the selected VPC.
Subnet IDs	`Requires input`	Select at least two subnets which have access to the OpenSearch domain. The log processor Lambda function will reside in the subnets. Make sure the subnets have access to the Amazon S3 service.

Parameter	Default	Description
Security Group ID	*Requires input*	Select a Security Group which will be associated with the log processor Lambda function. Make sure the Security Group has access to the OpenSearch domain.
S3 Backup Bucket	*Requires input*	The S3 backup bucket name to store the failed ingestion logs.
KMS-CMK ARN	<Optional>	The KMS-CMK ARN for encryption. Leave it blank to create a new KMS CMK.
Number Of Shards	5	Number of shards to distribute the index evenly across all data nodes. Keep the size of each shard between 10-50 GB.
Number of Replicas	1	Number of replicas for OpenSearch Index. Each replica is a full copy of an index.
Age to Warm Storage	<Optional>	The age required to move the index into warm storage (e.g. 7d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). This is only effective when warm storage is enabled in OpenSearch.

Parameter	Default	Description
Age to Cold Storage	<Optional>	The age required to move the index into cold storage (e.g. 30d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). This is only effective when cold storage is enabled in OpenSearch.
Age to Retain	<Optional>	The age to retain the index (e.g. 180d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). If value is "", the index will not be deleted.
Rollover Index Size	<Optional>	The minimum size of the shard storage required to roll over the index (e.g. 30GB).
Index Suffix	yyyy-MM-dd	The common suffix format of OpenSearch index for the log(Example: yyyy-MM-dd, yyyy-MM-dd-HH). The index name will be <Index Prefix>-<Log Type>-<Index Suffix>-000001.
Compression type	best_compression	The compression type to use to compress stored data. Available values are best_compression and default.

Parameter	Default	Description
Refresh Interval	1s	How often the index should refresh, which publishes its most recent changes and makes them available for searching. Can be set to -1 to disable refreshing. Default is 1s.
Plugins	<Optional>	List of plugins delimited by comma. Leave it blank if there are no available plugins to use. Valid inputs are user_agent, geo_ip.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template creates AWS Identity and Access Management (IAM) resources.
9. Choose **Create** stack to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately 10 minutes.

View dashboard

The dashboard includes the following visualizations.

Visualization Name	Source Field	Description
Total Requests	• log event	Displays aggregated events based on a specified time interval.

Visualization Name	Source Field	Description
Request History	• log event	Presents a bar chart that displays the distribution of events over time.
Request By Target	• log event • target_ip	Presents a bar chart that displays the distribution of events over time and IP.
Unique Visitors	• client_ip	Displays unique visitors identified by client IP address.
Status Code	• elb_status_code	Displays the count of requests made to the ALB, grouped by HTTP status codes (e.g., 200, 404, 403, etc.).
Status History	• elb_status_code	Shows the historical trend of HTTP status codes returned by the ALB over a specific period of time.
Status Code Pipe	• elb_status_code	Represents the distribution of requests based on different HTTP status codes using a pie chart.
Average Processing Time	• request_processing_time • response_processing_time • target_processing_time	This visualization calculate s and presents the average time taken for various operations in the ALB.
Avg. Processing Time History	• request_processing_time • response_processing_time • target_processing_time	Displays the historical trend of the average time-cons uming of each operation returned by the ALB within a specific period of time.

Visualization Name	Source Field	Description
Request Verb	• request_verb	Displays the count of requests made to the ALB using a pie chart, grouped by http request method names (e.g., POST, GET, HEAD, etc.).
Total Bytes	• received_bytes • sent_bytes	Provides insights into data transfer activities, including the total bytes transferred.
Sent and Received Bytes History	• received_bytes • sent_bytes	Displays the historical trend of the the received bytes, send bytes
SSL Protocol	• ssl_protocol	Displays the count of requests made to the ALB, grouped by SSL Protocol
Top Request URLs	• request_url	The web requests view enables you to analyze the top web requests.
Top Client IPs	• client_ip	Provides the top 10 IP address accessing your ALB.
Top User Agents	• user_agent	Provides the top 10 user agents accessing your ALB.
Target Status	• target_ip • target_status_code	Displays the http status code request count for targets in ALB target group.

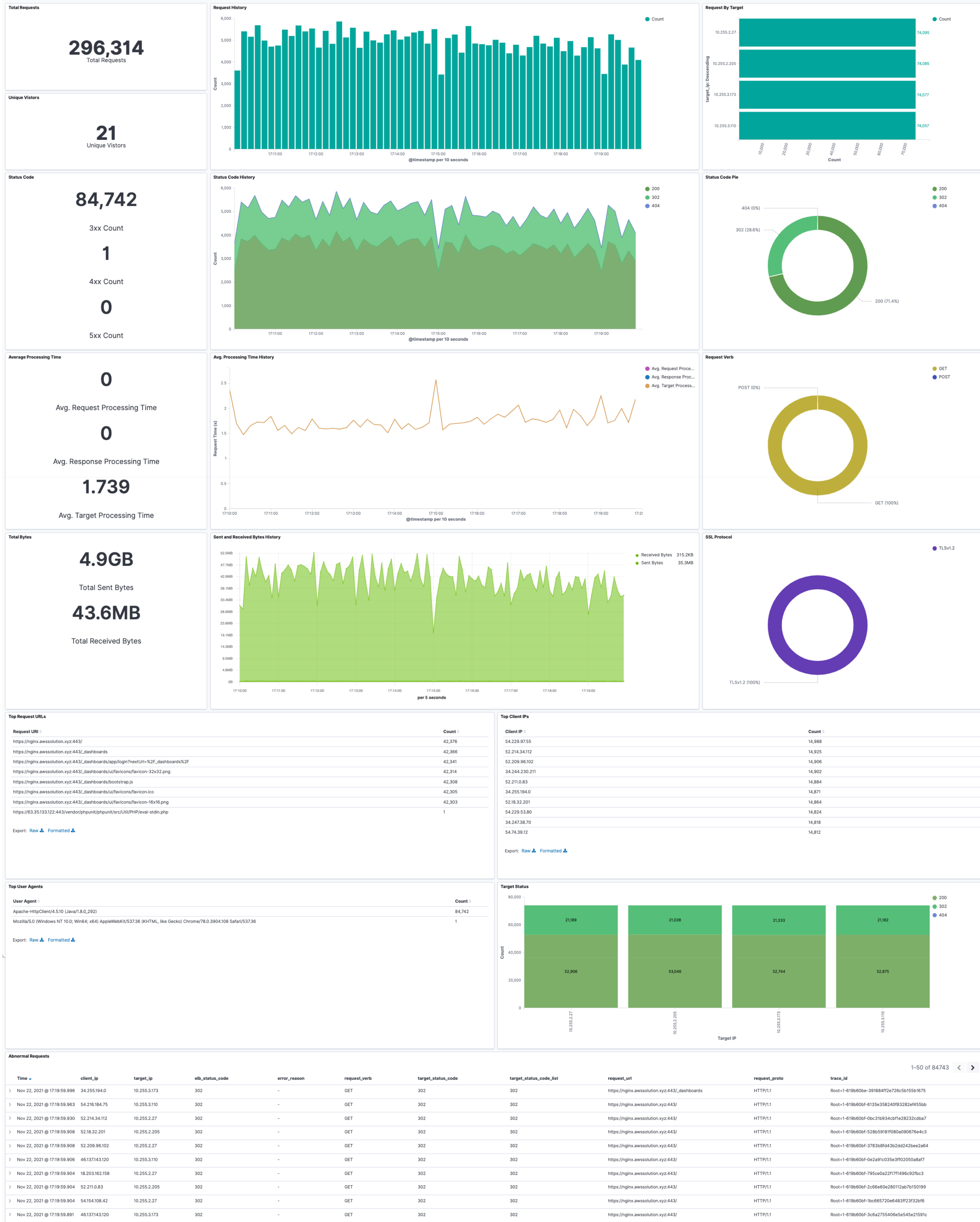
Visualization Name	Source Field	Description
Abnormal Requests	• @timestamp • client_ip • target_ip • elb_status_code • error_reason • request_verb • target_status_code • target_status_code_list • request_url • request_proto • trace_id	Provides a detailed list of log events, including timestamps, client ip, target ip, etc.
Requests by OS	• ua_os	Displays the count of requests made to the ALB, grouped by user agent OS
Request by Device	• ua_device	Displays the count of requests made to the ALB, grouped by user agent device.
Request by Browser	• ua_browser	Displays the count of requests made to the ALB, grouped by user agent browser.
Request by Category	• ua_category	Displays the count of category made to the ALB, grouped by user agent category (e.g., PC, Mobile, Tablet, etc.).
Requests by Countries or Regions	• geo_iso_code	Displays the count of requests made to the ALB (grouped by the corresponding country or region resolved by the client IP).

Visualization Name	Source Field	Description
Top Countries or Regions	• geo_country	Top 10 countries with the ALB Access.
Top Cities	• geo_city	Top 10 cities with ALB Access

Sample Dashboard

You can access the built-in dashboard in Amazon OpenSearch Service to view log data. For more information, see Access Dashboard.

You can click the below image to view the high-resolution sample dashboard.


Total Requests
296,314
Total Requests
Request History
Request By Target
Unique Vistors
21
Unique Vistors
Status Code
84,742
3xx Count
1
4xx Count
0
5xx Count
Status Code History
Status Code Pie
Average Processing Time
0
Avg. Request Processing Time
0
Avg. Response Processing Time
1.739
Avg. Target Processing Time
Avg. Processing Time History
Request Verb
Total Bytes
4.9GB
Total Sent Bytes
43.6MB
Total Received Bytes
Sent and Received Bytes History
SSL Protocol
Top Request URLs
Request URI
Count
https://nginx.awssolution.xyz:443/ 42,376
https://nginx.awssolution.xyz:443/_dashboards 42,366
https://nginx.awssolution.xyz:443/_dashboards/app/login?nextUrl=%2F_dashboards%2F 42,341
https://nginx.awssolution.xyz:443/_dashboards/ui/favicons/favicon-32x32.png 42,314
https://nginx.awssolution.xyz:443/_dashboards/bootstrap.js 42,308
https://nginx.awssolution.xyz:443/_dashboards/ui/favicons/favicon.ico 42,305
https://nginx.awssolution.xyz:443/_dashboards/ui/favicons/favicon-16x16.png 42,303
https://63.35.133.122:443/vendor/phpunit/phpunit/src/Util/PHP/eval-stdin.php 1
Export: Raw Formatted
Top Client IPs
Client IP
Count
54.229.97.55 14,988
52.214.34.112 14,925
52.209.96.102 14,906
34.244.230.211 14,902
52.211.0.83 14,884
34.255.194.0 14,871
52.18.32.201 14,864
54.229.53.80 14,824
34.247.38.70 14,818
54.74.39.12 14,812
Export: Raw Formatted
Top User Agents
Target Status
Abnormal Requests

Create log ingestion (Light Engine for log analytics)

Using the Centralized Logging with OpenSearch Console

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the navigation pane, under **Log Analytics Pipelines**, choose **Service Log**.
3. Choose the **Create a log ingestion** button.
4. In the **AWS Services** section, choose **Elastic Load Balancer**.
5. Choose **Light Engine**, and choose **Next**.
6. Under **Specify settings**, choose **Automatic** or **Manual** for **CloudFront logs enabling**. The automatic mode will detect the CloudFront log location automatically.
   - For **Automatic mode**, choose an application log balancer from the dropdown list. If the selected ALB access log is not enable, choose **Enable** to enable the ALB access log.
   - For **Manual mode**, enter the **Application Load Balancer identifier** and **Log location**.
   - (Optional) If you are ingesting CloudFront logs from another account, select a linked account from the **Account** dropdown list first.
7. Choose **Next**.
8. Choose **Log Processing Enriched fields** if needed. The available plugins are **location** and **OS/User Agent**. Enabling rich fields may increase data processing latency and processing cost. By default, it is not selected.
9. In the **Specify Light Engine Configuration** section, if you want to ingest an associated templated Grafana dashboard, select **Yes** for the sample dashboard.
10. You can choose an existing Grafana, or you can import a new one by making configurations in Grafana.
11. Select an Amazon S3 bucket to store partitioned logs and give a name to the log table. The solution provides a predefined table name, but you can modify it according to your needs.
12. Modify the log processing frequency if needed, which is set to **5** minutes by default with a minimum processing frequency of **1** minute.
13. In the **Log Lifecycle** section, if needed, enter the log merge time and log archive time to modify the default values provided by the solution.
14. Choose **Next**.
15. Add tags if needed.
16. Choose **Create**.

Using the CloudFormation Stack

This automated AWS CloudFormation template deploys the *Centralized Logging with OpenSearch - ALB Log Ingestion* solution in the AWS Cloud.

	Launch in AWS Management Console	Download Template
AWS Regions	Launch Stack	Template
AWS China Regions	Launch Stack	Template

1. Log in to the AWS Management Console and select above button to launch the AWS CloudFormation template. You can also download the template as a starting point for your own implementation.
2. To launch the stack in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following parameters.
    - Parameters for **Pipeline settings**

Parameter	Default	Description
Pipeline Id	`Requires input`	The unique identifier for the pipeline, which is essential if you need to create multiple ALB pipelines and write different ALB logs into separate tables. To ensure uniqueness, you can

Parameter	Default	Description
		generate a unique pipeline identifier using [uuidgenerator](#).
Staging Bucket Prefix	AWSLogs/ALBLogs	The storage directory for logs in the temporary storage area should ensure uniqueness and non-overlapping of the prefix for different pipelines.

- Parameters for **Destination settings**

Parameter	Default	Description
Centralized Bucket Name	*`Requires input`*	The name for the centralized S3 bucket. For example, `centralized-logging-bucket` .
Centralized Bucket Prefix	datalake	The centralized bucket prefix. By default, the database location is s3://{Centralized Bucket Name}/{Centralized Bucket Prefix}/amazon_cl_centralized.
Centralized Table Name	ALB	Table name for writing data to the centralized database. You can modify it if needed.

Parameter	Default	Description
Enrichment Plugins	<Optional input>	The available plugins to choose from are **location** and **OS/User Agent** . Enabling rich fields will increase data processin g latency and processing costs. It is not selected by default.

- Parameters for **Scheduler settings**

Parameter	Default	Description
LogProcessor Schedule Expression	rate (5 minutes)	Task scheduling expression for performing log processin g, with a default value of executing the LogProcessor every 5 minutes. For more information, see [Schedule types]().
LogMerger Schedule Expression	cron(0 1 * ?)	Task scheduling expression for performing log merging, with a default value of executing the LogMerger at 1 AM every day. For more information, see [Schedule types]().

Parameter	Default	Description
LogArchive Schedule Expression	cron(0 2 * ?)	Task scheduling expression for performing log archiving , with a default value of executing the LogArchive at 2 AM every day. For more information, see [Schedule types]().
Age to Merge	7	Small file retention days, with a default value of 7, indicating that logs older than 7 days will be merged into small files. It can be adjusted as needed.
Age to Archive	30	Log retention days, with a default value of 30, indicating that data older than 30 days will be archived and deleted. It can be adjusted as needed.

- Parameters for **Notification settings**

Parameter	Default	Description
Notification Service	SNS	Notification method for alerts. If your main stack is in AWS China Regions, you can only choose the SNS method. If your main stack is in AWS Regions, you can choose either the SNS or SES method.
Recipients	*Requires input*	If the Notification Service is SNS, enter the SNS Topic ARN to ensure that you have the required permissions. If the Notification Service is SES, enter the email addresses separated by commas to ensure that the email addresses are already Verified Identities in SES. The adminEmail provided during the creation of the main stack will receive a verification email by default.

- Parameters for **Dashboard settings**

Parameter	Default	Description
Import Dashboards	FALSE	Whether to import the Dashboard into Grafana. If it is set to `true`, you must provide the Grafana URL and Grafana Service Account Token.
Grafana URL	*`Requires input`*	Grafana access URL. For example, `https://alb-72277319.us-west-2.elb.amazonaws.com` .
Grafana Service Account Token	*`Requires input`*	Service Account Token created in Grafana.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template creates AWS Identity and Access Management (IAM) resources.
9. Choose **Create** stack to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately 10 minutes.

View dashboard

Visualization Name	Source Field	Description
Filters	Filters	The following data can be filtered by query filter conditions.
Total Requests	log event	Displays aggregated events based on a specified time interval.

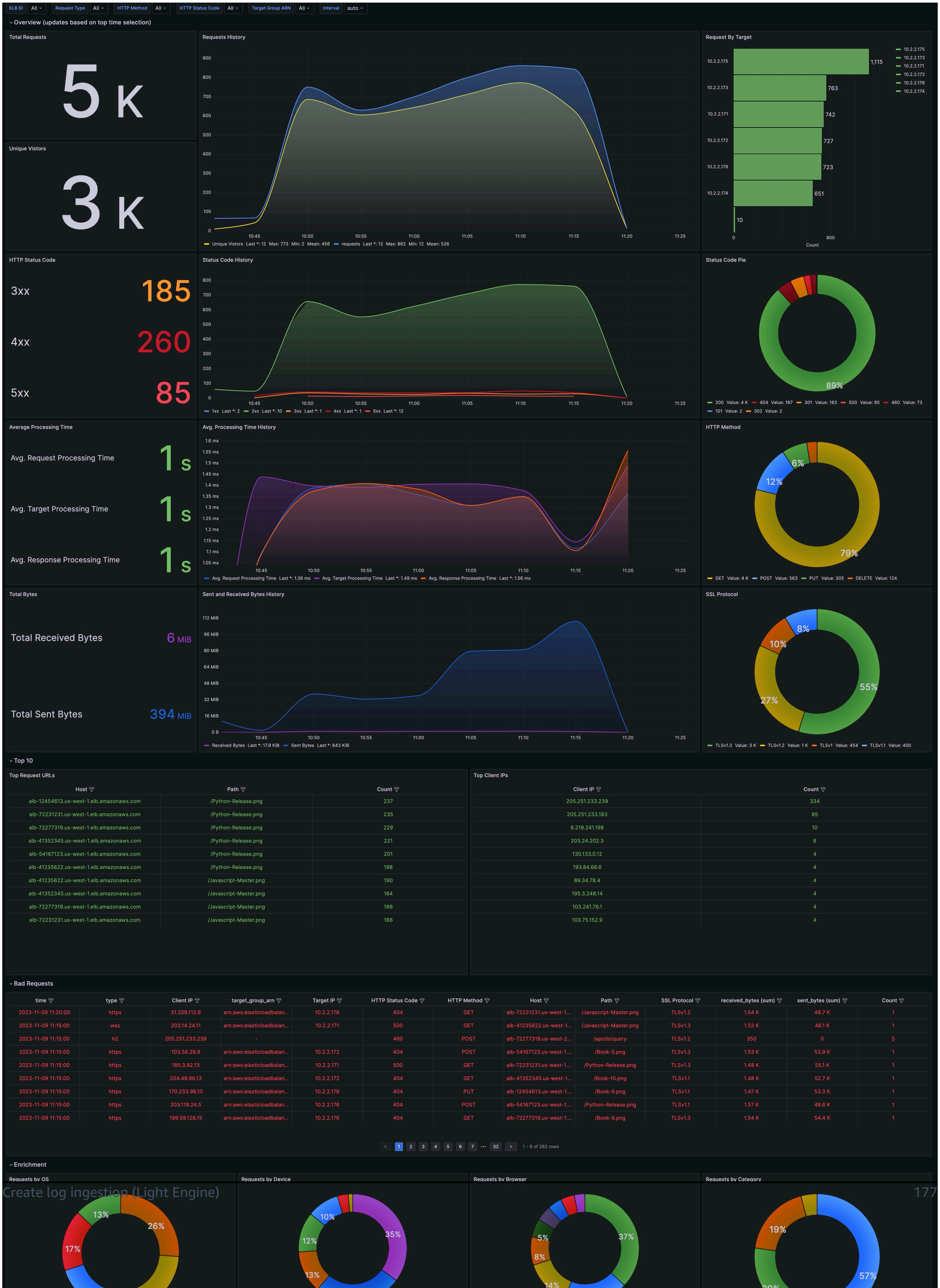
Visualization Name	Source Field	Description
Unique Visitors	client_ip	Displays unique visitors identified by client IP address.
Requests History	log event	Presents a bar chart that displays the distribution of events over time.
Request By Target	log event target_ip	Presents a bar chart that displays the distribution of events over time and IP.
HTTP Status Code	elb_status_code	Displays the count of requests made to the ALB, grouped by HTTP status codes (for example, 200, 404, 403).
Status Code History	elb_status_code	Shows the historical trend of HTTP status codes returned by the ALB over a specific period of time.
Status Code Pie	elb_status_code	Represents the distribution of requests based on different HTTP status codes using a pie chart.
Average Processing Time	request_processing_time response_processing_time target_processing_time	This visualization calculates and presents the average time taken for various operations in the ALB.
Avg. Processing Time History	request_processing_time response_processing_time target_processing_time	Displays the historical trend of the average time-consuming of each operation returned by the ALB within a specific period of time.
HTTP Method	request_verb	Displays the count of requests made to the ALB using a pie chart, grouped by HTTP request method names (for example, POST, GET, HEAD).

Visualization Name	Source Field	Description
Total Bytes	received_bytes sent_bytes	Provides insights into data transfer activities, including the total bytes transferred.
Sent and Received Bytes History	received_bytes sent_bytes	Displays the historical trend of the received bytes, send bytes.
SSL Protocol	ssl_protocol	Displays the count of requests made to the ALB, grouped by SSL Protocol.
Top Request URLs	request_url	The web requests view enables you to analyze the top web requests.
Top Client IPs	client_ip	Provides the top 10 IP addresses accessing your ALB.
Bad Requests	type client_ip target_group_arn target_ip elb_status_code request_verb request_url ssl_protocol received_bytes sent_bytes	Provides a detailed list of log events, including timestamps, client IP, target IP, etc.
Requests by OS	ua_os	Displays the count of requests made to the ALB, grouped by user agent OS.
Requests by Device	ua_device	Displays the count of requests made to the ALB, grouped by user agent device.
Requests by Browser	ua_browser	Displays the count of requests made to the ALB, grouped by user agent browser.

Visualization Name	Source Field	Description
Requests by Category	ua_category	Displays the count of category made to the ALB, grouped by user agent category (for example, PC, Mobile, Tablet).
Requests by Countries or Regions	geo_iso_code	Displays the count of requests made to the ALB (grouped by the corresponding country or region resolved by the client IP).
Top Countries or Regions	geo_country	Top 10 countries with the ALB Access.
Top Cities	geo_city	Top 10 cities with ALB Access.

Sample dashboard

Below shows the sample dashboard.


ELB ID All
Request Type All
HTTP Method All
HTTP Status Code All
Target Group ARN All
interval auto
Overview (updates based on top time selection)
Total Requests
5 K
Unique Vistors
3 K
Requests History
Unique Vistors Last *: 12 Max: 773 Min: 2 Mean: 456
requests Last *: 12 Max: 862 Min: 12 Mean: 526
Request By Target
10.2.2.175 1,115
10.2.2.173 763
10.2.2.171 742
10.2.2.172 727
10.2.2.176 723
10.2.2.174 651
10
Count
HTTP Status Code
3xx 185
4xx 260
5xx 85
Status Code History
1xx Last *: 2
2xx Last *: 10
3xx Last *: 1
4xx Last *: 1
5xx Last *: 12
Status Code Pie
89%
200 Value: 4 K
404 Value: 187
301 Value: 183
500 Value: 85
460 Value: 73
101 Value: 2
302 Value: 2
Average Processing Time
Avg. Request Processing Time 1 s
Avg. Target Processing Time 1 s
Avg. Response Processing Time 1 s
Avg. Processing Time History
Avg. Request Processing Time Last *: 1.36 ms
Avg. Target Processing Time Last *: 1.49 ms
Avg. Response Processing Time Last *: 1.56 ms
HTTP Method
6%
12%
79%
GET Value: 4 K
POST Value: 563
PUT Value: 305
DELETE Value: 124
Total Bytes
Total Received Bytes 6 MiB
Total Sent Bytes 394 MiB
Sent and Received Bytes History
Received Bytes Last *: 17.9 KiB
Sent Bytes Last *: 643 KiB
SSL Protocol
8%
10%
55%
27%
TLSv1.3 Value: 3 K
TLSv1.2 Value: 1 K
TLSv1 Value: 454
TLSv1.1 Value: 400
Top 10
Top Request URLs
Host Path Count
alb-12454613.us-west-1.elb.amazonaws.com /Python-Release.png 237
alb-72231231.us-west-1.elb.amazonaws.com /Python-Release.png 235
alb-72277319.us-west-1.elb.amazonaws.com /Python-Release.png 229
alb-41352345.us-west-1.elb.amazonaws.com /Python-Release.png 221
alb-54167123.us-west-1.elb.amazonaws.com /Python-Release.png 201
alb-41235622.us-west-1.elb.amazonaws.com /Python-Release.png 198
alb-41235622.us-west-1.elb.amazonaws.com /Javascript-Master.png 190
alb-41352345.us-west-1.elb.amazonaws.com /Javascript-Master.png 184
alb-72277319.us-west-1.elb.amazonaws.com /Javascript-Master.png 166
alb-72231231.us-west-1.elb.amazonaws.com /Javascript-Master.png 166
Top Client IPs
Client IP Count
205.251.233.239 334
205.251.233.183 85
8.219.241.198 10
203.24.202.3 6
130.133.0.12 4
193.84.66.6 4
89.34.78.4 4
195.3.248.14 4
103.241.76.1 4
103.75.152.9 4
Bad Requests
time type Client IP target_group_arn Target IP HTTP Status Code HTTP Method Host Path SSL Protocol received_bytes (sum) sent_bytes (sum) Count
2023-11-09 11:20:00 https 31.209.112.6 arn:aws:elasticloadbalan... 10.2.2.176 404 GET alb-72231231.us-west-1.... /Javascript-Master.png TLSv1.2 1.54 K 48.7 K 1
2023-11-09 11:15:00 wss 203.14.24.11 arn:aws:elasticloadbalan... 10.2.2.171 500 GET alb-41235622.us-west-1.... /Javascript-Master.png TLSv1.3 1.52 K 46.1 K 1
2023-11-09 11:15:00 h2 205.251.233.239 - 460 POST alb-72277319.us-west-2... /api/ds/query TLSv1.2 350 0 5
2023-11-09 11:15:00 https 103.56.28.9 arn:aws:elasticloadbalan... 10.2.2.172 404 POST alb-54167123.us-west-1.... /Book-5.png TLSv1.3 1.53 K 53.9 K 1
2023-11-09 11:15:00 https 185.3.92.13 arn:aws:elasticloadbalan... 10.2.2.171 500 GET alb-72231231.us-west-1.... /Python-Release.png TLSv1.3 1.48 K 55.1 K 1
2023-11-09 11:15:00 https 204.48.96.13 arn:aws:elasticloadbalan... 10.2.2.172 404 GET alb-41352345.us-west-1.... /Book-10.png TLSv1.1 1.48 K 52.7 K 1
2023-11-09 11:15:00 https 170.233.96.10 arn:aws:elasticloadbalan... 10.2.2.176 404 PUT alb-12454613.us-west-1.... /Book-9.png TLSv1.1 1.47 K 53.3 K 1
2023-11-09 11:15:00 https 203.119.24.5 arn:aws:elasticloadbalan... 10.2.2.176 404 POST alb-54167123.us-west-1.... /Python-Release.png TLSv1.1 1.57 K 48.6 K 1
2023-11-09 11:15:00 https 199.59.128.15 arn:aws:elasticloadbalan... 10.2.2.176 404 GET alb-72277319.us-west-1.... /Book-9.png TLSv1.3 1.54 K 54.4 K 1
1 2 3 4 5 6 7 ... 32
1 - 9 of 282 rows
Enrichment
Requests by OS
13%
26%
17%
Requests by Device
10%
35%
12%
13%
Requests by Browser
5%
37%
8%
14%
Requests by Category
19%
57%

AWS WAF Logs

WAF Access logs provide detailed information about traffic that is analyzed by your web ACL. Logged information includes the time that AWS WAF received a web request from your AWS resource, detailed information about the request, and details about the rules that the request matched.

You can create a log ingestion into Amazon OpenSearch Service or Light Engine either by using the Centralized Logging with OpenSearch console or by deploying a standalone CloudFormation stack.

Important

- You must deploy Centralized Logging with OpenSearch solution in the same region as your Web ACLs, or you will not be able to create a WAF pipeline. For example:
  - If your Web ACL is associated with Global Cloudfront, you must deploy the solution in us-east-1.
  - If your Web ACL is associated with other resources in regions like Ohio, your Centralized Logging with OpenSearch stack must also be deployed in that region.
- The WAF logging bucket must be the same as the Centralized Logging with OpenSearch solution.
- WAF Classic logs are not supported in Centralized Logging with OpenSearch. Learn more about migrating rules from WAF Classic to the new AWS WAF.
- The Amazon OpenSearch Service index is rotated on a daily basis by default, and you can adjust the index in the Additional Settings.

Create log ingestion (Amazon OpenSearch Service for log analytics)

Using the Centralized Logging with OpenSearch Console

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the navigation pane, under **Log Analytics Pipelines**, choose **Service Log**.
3. Choose the **Create a log ingestion** button.
4. In the **AWS Services** section, choose **AWS WAF**.
5. Choose **Next**.
6. Under **Specify settings**, choose **Automatic** or **Manual**.
   - For **Automatic** mode, choose a Web ACL in the dropdown list.

- For **Manual** mode, enter the **Web ACL name**.
- (Optional) If you are ingesting WAF logs from another account, select a linked account from the **Account** dropdown list first.

7. Specify an **Ingest Options**. Choose between **Sampled Request** or **Full Request**.
   - For **Sampled Request**, enter how often you want to ingest sampled requests in minutes.
   - For **Full Request**, if the Web ACL log is not enabled, choose **Enable** to enable the access log, or enter **Log location** in Manual mode. Note that Centralized Logging with OpenSearch will automatically enable logging with a Firehose stream as destination for your WAF.
8. Choose **Next**.
9. In the **Specify OpenSearch domain** section, select an imported domain for **Amazon OpenSearch Service domain**.
10. Choose **Yes** for **Sample dashboard** if you want to ingest an associated templated Amazon OpenSearch Service dashboard.
11. You can change the **Index Prefix** of the target Amazon OpenSearch Service index if needed. The default prefix is the Web ACL Name.
12. In the **Log Lifecycle** section, enter the number of days to manage the Amazon OpenSearch Service index lifecycle. The Centralized Logging with OpenSearch will create the associated Index State Management (ISM) policy automatically for this pipeline.
13. In the **Select log processor** section, choose the log processor.
    - When selecting Lambda as log processor, you can configure the Lambda concurrency if needed.
    - (Optional) OSI as log processor is now supported in these Regions. When OSI is selected, enter the minimum and maximum number of OCU. For more information, see Scaling pipelines.
14. Choose **Next**.
15. Add tags if needed.
16. Choose **Create**.

Using the CloudFormation Stack

This automated AWS CloudFormation template deploys the *Centralized Logging with OpenSearch - WAF Log Ingestion* solution in the AWS Cloud.

	Launch in AWS Management Console	Download Template
AWS Regions (Full Request)	Launch Stack	Template
AWS China Regions (Full Request)	Launch Stack	Template
AWS Regions (Sampled Request)	Launch Stack	Template
AWS China Regions (Sampled Request)	Launch Stack	Template

1. Log in to the AWS Management Console and select the button to launch the AWS CloudFormation template. You can also download the template as a starting point for your own implementation.
2. To launch the stack in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following parameters.
   - Parameters for **Full Request** only

Parameter	Default	Description
Log Bucket Name	`Requires input`	The S3 bucket name which stores the logs.
Log Bucket Prefix	`Requires input`	The S3 bucket path prefix which stores the logs.

   - Parameters for **Sampled Request** only

Parameter	Default	Description
WebACL Names	*`Requires input`*	The list of Web ACL names, delimited by comma.
Interval	2	The default interval (in minutes) to get sampled logs. The value must be between 2 and 180.

- Common parameters

Parameter	Default	Description
Log Source Account ID	<Optional>	The AWS Account ID of the S3 bucket. Required for cross-account log ingestion (Please add a member account first). By default, the Account ID you logged in at **Step 1** will be used.
Log Source Region	<Optional>	The AWS Region of the S3 bucket. By default, the Region you selected at **Step 2** will be used.
Log Source Account Assume Role	<Optional>	The IAM Role ARN used for cross-account log ingestion . Required for cross-account log ingestion (Please add a member account first).
Engine Type	OpenSearch	The engine type of the OpenSearch. Select OpenSearch.

Parameter	Default	Description
OpenSearch Domain Name	*Requires input*	The domain name of the Amazon OpenSearch Service cluster.
OpenSearch Endpoint	*Requires input*	The OpenSearch endpoint URL. For example, vpc-your_opensearch_domain_name-xcvgw6uu2o6zafsiefxubwuohe.us-east-1.es.amazonaws.com
Index Prefix	*Requires input*	The common prefix of OpenSearch index for the log. The index name will be <Index Prefix>-<log-type>-<YYYY-MM-DD>.
Create Sample Dashboard	Yes	Whether to create a sample OpenSearch dashboard.
VPC ID	*Requires input*	Select a VPC which has access to the OpenSearch domain. The log processor Lambda function will reside in the selected VPC.
Subnet IDs	*Requires input*	Select at least two subnets which have access to the OpenSearch domain. The log processor Lambda function will reside in the subnets. Make sure the subnets have access to the Amazon S3 service.

Parameter	Default	Description
Security Group ID	*Requires input*	Select a Security Group which will be associate d with the log processor Lambda function. Make sure the Security Group has access to the OpenSearch domain.
S3 Backup Bucket	*Requires input*	The S3 backup bucket name to store the failed ingestion logs.
KMS-CMK ARN	<Optional input>	The KMS-CMK ARN for encryption. Leave it blank to create a new KMS CMK.
Number Of Shards	5	Number of shards to distribute the index evenly across all data nodes. Keep the size of each shard between 10-50 GB.
Number of Replicas	1	Number of replicas for OpenSearch Index. Each replica is a full copy of an index.
Days to Warm Storage	0	The number of days required to move the index into warm storage. This takes effect only when the value is larger than 0 and warm storage is enabled in OpenSearch.

Parameter	Default	Description
Days to Cold Storage	0	The number of days required to move the index into cold storage. This takes effect only when the value is larger than 0 and cold storage is enabled in OpenSearch.
Days to Retain	0	The total number of days to retain the index. If value is 0, the index will not be deleted.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template creates AWS Identity and Access Management (IAM) resources.
9. Choose **Create** stack to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately 10 minutes.

View dashboard

The dashboard includes the following visualizations.

Visualization Name	Source Field	Description
Filters	• Filters	The following data can be filtered by query filter conditions.
Web ACLs	• log event • webaclName	Displays the count of requests made to the WAF, grouped by Web ACL Names.

Visualization Name	Source Field	Description
Total Requests	• log event	Displays the total number of web requests.
Request Timeline	• log event	Presents a bar chart that displays the distribution of events over time.
WAF Rules	• terminatingRuleId	Presents a pie chart that displays the distribution of events over the WAF rules in the Web ACL.
Total Blocked Requests	• log event	Displays the total number of blocked web requests.
Unique Client IPs	• Request.ClientIP	Displays unique visitors identified by client IP.
Country or Region By Request	• Request.Country	Displays the count of requests made to the Web ACL (grouped by the correspon ding country or region resolved by the client IP).
Http Methods	• Request.HTTPMethod	Displays the count of requests made to the Web ACL using a pie chart, grouped by http request method names (e.g., POST, GET, HEAD, etc.).
Http Versions	• Request.HTTPVersion	Displays the count of requests made to the Web ACL using a pie chart, grouped by http protocol version (e.g., HTTP/2.0, HTTP/1.1, etc.).

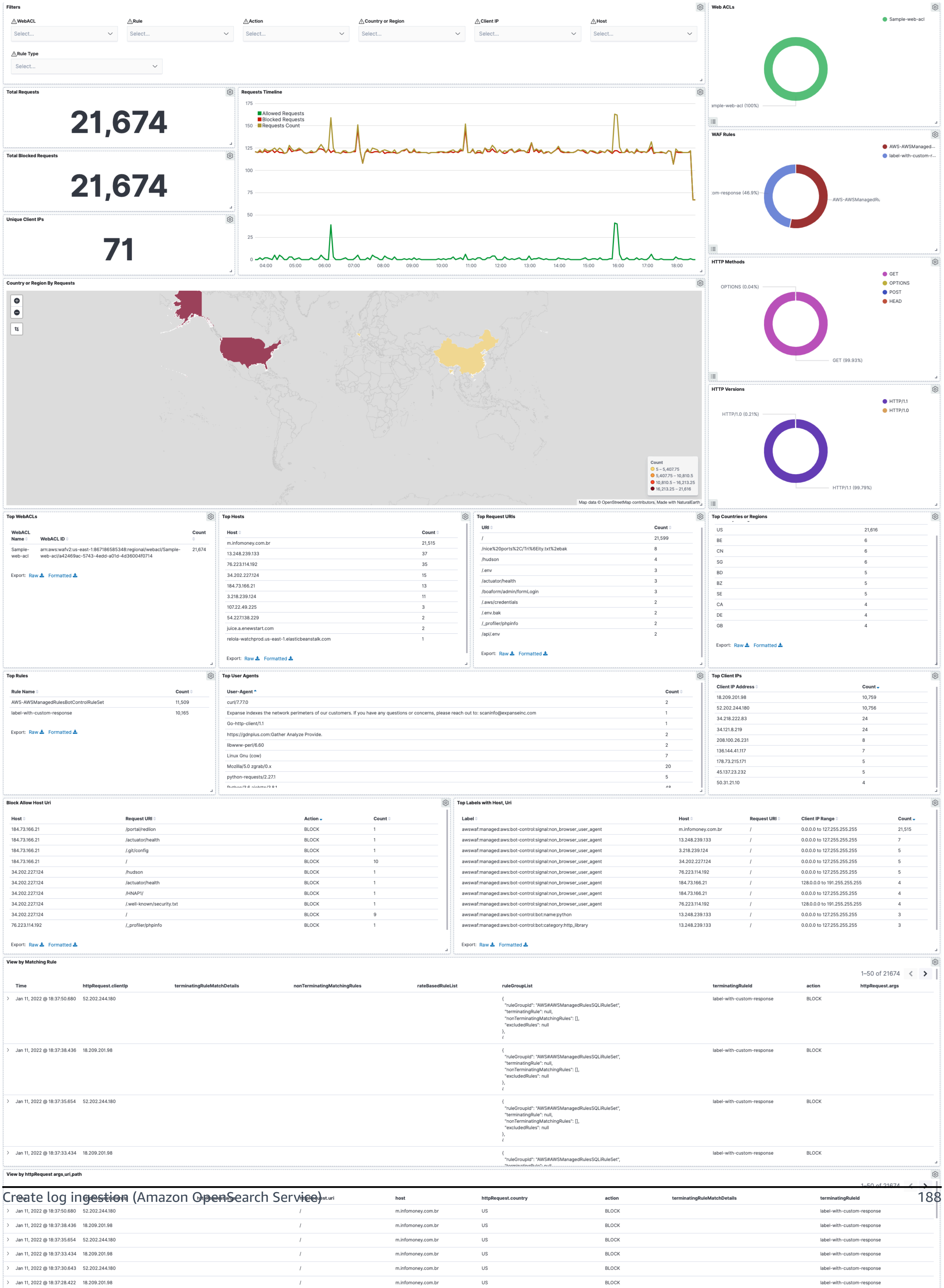
Visualization Name	Source Field	Description
Top WebACLs	• webaclName • webaclId.keyword	The web requests view enables you to analyze the top web requests.
Top Hosts	• host	Lists the source IP addresses associated with events, enabling you to identify and investigate potentially suspicious or unauthorized activities.
Top Request URIs	• Request.URI	Top 10 request URIs.
Top Countries or Regions	• Request.country	Top 10 countries with the Web ACL Access.
Top Rules	• terminatingRuleId	Top 10 rules in the web ACL that matched the request.
Top Client IPs	• Request.ClientIP	Provides the top 10 IP address.
Top User Agents	• userAgent	Provides the top 10 user agents
Block Allow Host Uri	• host • Request.URI • action	Provides blocked or allowed web requests.
Top Labels with Host, Uri	• labels.name • host • Request.URI	Top 10 detailed logs by labels with host, URI
View by Matching Rule	• sc-status	This visualization provides detailed logs by DQL "terminatingRuleId:*".

Visualization Name	Source Field	Description
View by httpRequest args,uri, path	• sc-status	This visualization provides detailed logs by DQL.

Sample Dashboard

You can access the built-in dashboard in Amazon OpenSearch Service to view log data. For more information, see Access Dashboard.

You can click the below image to view the high-resolution sample dashboard.


Filters
WebACL
Rule
Action
Country or Region
Client IP
Host
Rule Type
Select...
Web ACLs
Sample-web-acl
Total Requests
21,674
Total Blocked Requests
21,674
Unique Client IPs
71
Requests Timeline
Allowed Requests
Blocked Requests
Requests Count
WAF Rules
AWS-AWSManaged...
label-with-custom-r...
HTTP Methods
GET
OPTIONS
POST
HEAD
OPTIONS (0.04%)
GET (99.93%)
HTTP Versions
HTTP/1.1
HTTP/1.0
HTTP/1.0 (0.21%)
HTTP/1.1 (99.79%)
Country or Region By Requests
Count
5 – 5,407.75
5,407.75 – 10,810.5
10,810.5 – 16,213.25
16,213.25 – 21,616
Map data © OpenStreetMap contributors, Made with NaturalEarth
Top WebACLs
WebACL Name
WebACL ID
Count
Sample-web-acl
arn:aws:wafv2:us-east-1:867186585348:regional/webacl/Sample-web-acl/a42469ac-5743-4edd-a01d-4d36004f0714
21,674
Export: Raw Formatted
Top Hosts
Host
Count
m.infomoney.com.br 21,515
13.248.239.133 37
76.223.114.192 35
34.202.227.124 15
184.73.166.21 13
3.218.239.124 11
107.22.49.225 3
54.227.138.229 2
juice.a.enewstart.com 2
relola-watchprod.us-east-1.elasticbeanstalk.com 1
Top Request URIs
URI
/ 21,599
/nice%20ports%2C/Tri%6Eity.txt%2ebak 8
/hudson 4
/.env 3
/actuator/health 3
/boaform/admin/formLogin 3
/.aws/credentials 2
/.env.bak 2
/_profiler/phpinfo 2
/api/.env 2
Top Countries or Regions
US 21,616
BE 6
CN 6
SG 6
BD 5
BZ 5
SE 5
CA 4
DE 4
GB 4
Top Rules
Rule Name
AWS-AWSManagedRulesBotControlRuleSet 11,509
label-with-custom-response 10,165
Top User Agents
User-Agent
curl/7.77.0 2
Expanse indexes the network perimeters of our customers. If you have any questions or concerns, please reach out to: scaninfo@expanseinc.com 1
Go-http-client/1.1 1
https://gdnplus.com:Gather Analyze Provide. 2
libwww-perl/6.60 2
Linux Gnu (cow) 7
Mozilla/5.0 zgrab/0.x 20
python-requests/2.27.1 5
Top Client IPs
Client IP Address
18.209.201.98 10,759
52.202.244.180 10,756
34.218.222.83 24
34.121.8.219 24
208.100.26.231 8
136.144.41.117 7
178.73.215.171 5
45.137.23.232 5
50.31.21.10 4
Block Allow Host Uri
Host Request URI Action Count
184.73.166.21 /portal/redlion BLOCK 1
184.73.166.21 /actuator/health BLOCK 1
184.73.166.21 /.git/config BLOCK 1
184.73.166.21 / BLOCK 10
34.202.227.124 /hudson BLOCK 1
34.202.227.124 /actuator/health BLOCK 1
34.202.227.124 /HNAP1/ BLOCK 1
34.202.227.124 /.well-known/security.txt BLOCK 1
34.202.227.124 / BLOCK 9
76.223.114.192 /_profiler/phpinfo BLOCK 1
Top Labels with Host, Uri
Label Host Request URI Client IP Range Count
awswaf:managed:aws:bot-control:signal:non_browser_user_agent m.infomoney.com.br / 0.0.0.0 to 127.255.255.255 21,515
awswaf:managed:aws:bot-control:signal:non_browser_user_agent 13.248.239.133 / 0.0.0.0 to 127.255.255.255 7
awswaf:managed:aws:bot-control:signal:non_browser_user_agent 3.218.239.124 / 0.0.0.0 to 127.255.255.255 5
awswaf:managed:aws:bot-control:signal:non_browser_user_agent 34.202.227.124 / 0.0.0.0 to 127.255.255.255 5
awswaf:managed:aws:bot-control:signal:non_browser_user_agent 76.223.114.192 / 0.0.0.0 to 127.255.255.255 5
awswaf:managed:aws:bot-control:signal:non_browser_user_agent 184.73.166.21 / 128.0.0.0 to 191.255.255.255 4
awswaf:managed:aws:bot-control:signal:non_browser_user_agent 184.73.166.21 / 0.0.0.0 to 127.255.255.255 4
awswaf:managed:aws:bot-control:signal:non_browser_user_agent 76.223.114.192 / 128.0.0.0 to 191.255.255.255 4
awswaf:managed:aws:bot-control:bot:name:python 13.248.239.133 / 0.0.0.0 to 127.255.255.255 3
awswaf:managed:aws:bot-control:bot:category:http_library 13.248.239.133 / 0.0.0.0 to 127.255.255.255 3
View by Matching Rule
1–50 of 21674
Time httpRequest.clientIp terminatingRuleMatchDetails nonTerminatingMatchingRules rateBasedRuleList ruleGroupList terminatingRuleId action httpRequest.args
Jan 11, 2022 @ 18:37:50.680 52.202.244.180 label-with-custom-response BLOCK
Jan 11, 2022 @ 18:37:38.436 18.209.201.98 label-with-custom-response BLOCK
Jan 11, 2022 @ 18:37:35.654 52.202.244.180 label-with-custom-response BLOCK
Jan 11, 2022 @ 18:37:33.434 18.209.201.98 label-with-custom-response BLOCK
"ruleGroupId": "AWS#AWSManagedRulesSQLiRuleSet",
"terminatingRule": null,
"nonTerminatingMatchingRules": [],
"excludedRules": null
View by httpRequest args,uri,path
host httpRequest.country action terminatingRuleMatchDetails terminatingRuleId
Jan 11, 2022 @ 18:37:50.680 52.202.244.180 / m.infomoney.com.br US BLOCK label-with-custom-response
Jan 11, 2022 @ 18:37:38.436 18.209.201.98 / m.infomoney.com.br US BLOCK label-with-custom-response
Jan 11, 2022 @ 18:37:35.654 52.202.244.180 / m.infomoney.com.br US BLOCK label-with-custom-response
Jan 11, 2022 @ 18:37:33.434 18.209.201.98 / m.infomoney.com.br US BLOCK label-with-custom-response
Jan 11, 2022 @ 18:37:30.643 52.202.244.180 / m.infomoney.com.br US BLOCK label-with-custom-response
Jan 11, 2022 @ 18:37:28.422 18.209.201.98 / m.infomoney.com.br US BLOCK label-with-custom-response

Create log ingestion (Light Engine for log analytics)

Using the Centralized Logging with OpenSearch Console

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the navigation pane, under **Log Analytics Pipelines**, choose **Service Log**.
3. Choose the **Create a log ingestion** button.
4. In the **AWS Services** section, choose **AWS WAF**.
5. Choose **Light Engine**, and choose **Next**.
6. Under **Specify settings**, choose **Automatic** or **Manual**.
    - For **Automatic mode**, choose a Web ACL from the drop-down list.
    - For **Manual mode**, enter the Web ACL name.
7. (Optional) If you need to ingest logs across AWS accounts, select a linked account from the **Account** drop-down list first.
8. In the **Ingestion Options** section, select **Full Request**. If Web ACL is not enabled, choose **Enable Access Logging** to enable access logs. Alternatively, enter the log location in manual mode. Note that using the log delivery stream will automatically enable using Kinesis Data Firehose as the target for WAF logs.
9. Choose **Next**.
10. In the **Specify Light Engine Configuration** section, if you want to ingest an associated templated Grafana dashboard, select **Yes** for the sample dashboard.
11. You can choose an existing Grafana, or you can import a new one by making configurations in Grafana.
12. Select an Amazon S3 bucket to store partitioned logs and give a name to the log table. The solution provides a predefined table name, but you can modify it according to your needs.
13. Modify the log processing frequency if needed, which is set to **5** minutes by default with a minimum processing frequency of **1** minute.
14. In the **Log Lifecycle** section, if needed, enter the log merge time and log archive time to modify the default values provided by the solution.
15. Choose **Next**.
16. Add tags if needed.
17. Choose **Create**.

Using the CloudFormation Stack

This automated AWS CloudFormation template deploys the *Centralized Logging with OpenSearch - WAF Log Ingestion* solution in the AWS Cloud.

	Launch in AWS Management Console	Download Template
AWS Regions	Launch Stack	Template
AWS China Regions	Launch Stack	Template

1. Log in to the AWS Management Console and select above button to launch the AWS CloudFormation template. You can also download the template as a starting point for your own implementation.
2. To launch the stack in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following parameters.
   - Parameters for **Pipeline settings**

Parameter	Default	Description
Pipeline Id	*Requires input*	The unique identifier for the pipeline, which is essential if you need to create multiple WAF pipelines and write different WAF logs into separate tables. To ensure uniqueness, you can generate a unique pipeline identifier using uuidgenerator.

Parameter	Default	Description
Staging Bucket Prefix	AWSLogs/WAFLogs	The storage directory for logs in the temporary storage area should ensure uniqueness and non-overlapping of the prefix for different pipelines.

- Parameters for **Destination settings**

Parameter	Default	Description
Centralized Bucket Name	*`Requires input`*	The name for the centralized S3 bucket. For example, `centralized-logging-bucket` .
Centralized Bucket Prefix	datalake	The centralized bucket prefix. By default, the database location is s3://{Centralized Bucket Name}/{Centralized Bucket Prefix}/amazon_cl_centralized.
Centralized Table Name	WAF	Table name for writing data to the centralized database. You can modify it if needed.

- Parameters for **Scheduler settings**

Parameter	Default	Description
LogProcessor Schedule Expression	rate (5 minutes)	Task scheduling expression for performing log processin g, with a default value of executing the LogProcessor every 5 minutes. For more information, see Schedule types.
LogMerger Schedule Expression	cron(0 1 * ?)	Task scheduling expression for performing log merging, with a default value of executing the LogMerger at 1 AM every day. For more information, see Schedule types.
LogArchive Schedule Expression	cron(0 2 * ?)	Task scheduling expression for performing log archiving , with a default value of executing the LogArchive at 2 AM every day. For more information, see Schedule types.
Age to Merge	7	Small file retention days, with a default value of 7, indicating that logs older than 7 days will be merged into small files. It can be adjusted as needed.

Parameter	Default	Description
Age to Archive	30	Log retention days, with a default value of 30, indicating that data older than 30 days will be archived and deleted. It can be adjusted as needed.

- Parameters for **Notification settings**

Parameter	Default	Description
Notification Service	SNS	Notification method for alerts. If your main stack is in AWS China Regions, you can only choose the SNS method. If your main stack is in AWS Regions, you can choose either the SNS or SES method.

Parameter	Default	Description
Recipients	*`Requires input`*	If the Notification Service is SNS, enter the SNS Topic ARN to ensure that you have the required permissions. If the Notification Service is SES, enter the email addresses separated by commas to ensure that the email addresses are already Verified Identities in SES. The adminEmail provided during the creation of the main stack will receive a verification email by default.

- Parameters for **Dashboard settings**

Parameter	Default	Description
Import Dashboards	FALSE	Whether to import the Dashboard into Grafana. If it is set to `true`, you must provide the Grafana URL and Grafana Service Account Token.
Grafana URL	*`Requires input`*	Grafana access URL. For example, `https://alb-72277319.us-west-2.elb.amazonaws.com` .
Grafana Service Account Token	*`Requires input`*	Service Account Token created in Grafana.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template creates AWS Identity and Access Management (IAM) resources.
9. Choose **Create** stack to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately 10 minutes.

View dashboard

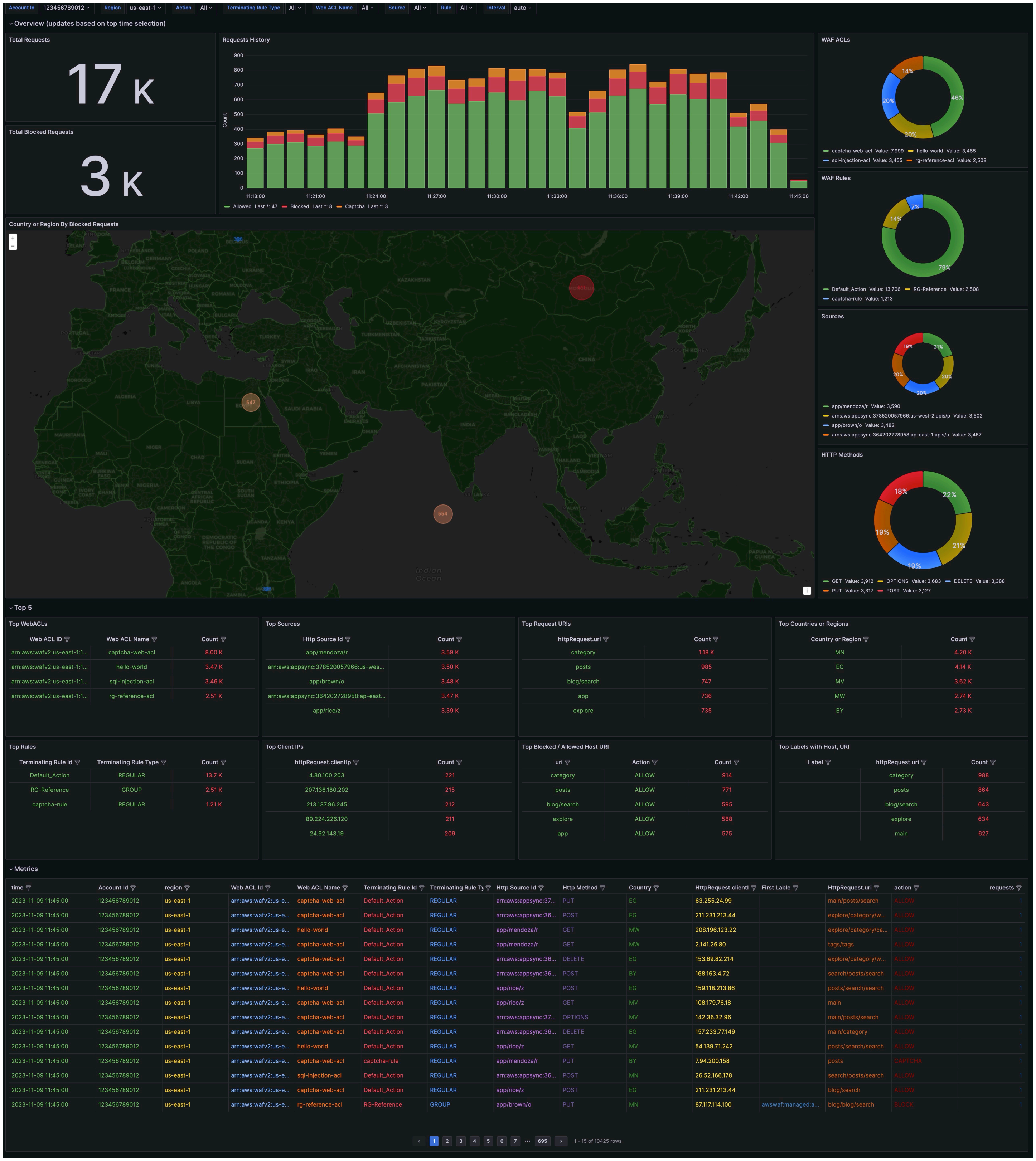
Visualization Name	Source Field	Description
Filters	Filters	The following data can be filtered by query filter conditions.
Total Requests	log event	Displays the total number of web requests.
Total Blocked Requests	log event	Displays the total number of blocked web requests.
Requests History	log event	Presents a bar chart that displays the distribution of events over time.
WAF ACLs	log event webaclName	Displays the count of requests made to the WAF, grouped by Web ACL Names.
WAF Rules	terminatingRuleId	Presents a pie chart that displays the distribution of events over the WAF rules in the Web ACL.

Visualization Name	Source Field	Description
Sources	httpSourceId	Presents a pie chart that displays the distribution of events over the id of the associated resource.
HTTP Methods	httpRequest.HTTPMethod	Displays the count of requests made to the Web ACL using a pie chart, grouped by HTTP request method names (for example, POST, GET, HEAD).
Country or Region By Blocked Requests	HTTPRequest.Country	Displays the count of blocked web requests made to the Web ACL (grouped by the corresponding country or region resolved by the client IP).
Top WebACLs	webaclName	The web requests view enables you to analyze the top web requests.
Top Sources	httpSourceId	Top 10 id of the associated resource.
Top Requests URIs	httpRequest.URI	Top 10 request URIs.
Top Countries or Regions	httpRequest.country	Top 10 countries with the Web ACL Access.
Top Rules	terminatingRuleId	Top 10 rules in the web ACL that matched the request.
Top Client IPs	httpRequest.ClientIP	Provides the top 10 IP addresses.

Visualization Name	Source Field	Description
Top Blocked / Allowed Hosts URI	host httpRequest.URI action	Provides blocked or allowed web requests.
Top Labels with Host, URI	labels host httpRequest.URI	Top 10 detailed logs by labels with host, URI.
Metrics	webaclId webaclName terminatingRuleId terminatingRuleType httpSourceId httpRequest.HTTPMethod httpReque st.country httpRequest.ClientIP labels httpReque st.URI action	Provides a detailed list of log events, including timestamps, WebACL, client IP and so on.

Sample Dashboard

Below shows the sample dashboard.


Account Id 123456789012
Region us-east-1
Action All
Terminating Rule Type All
Web ACL Name All
Source All
Rule All
Interval auto
Overview (updates based on top time selection)
Total Requests
17 K
Total Blocked Requests
3 K
Requests History
Allowed Last *: 47
Blocked Last *: 8
Captcha Last *: 3
WAF ACLs
captcha-web-acl Value: 7,999
hello-world Value: 3,465
sql-injection-acl Value: 3,455
rg-reference-acl Value: 2,508
WAF Rules
Default_Action Value: 13,706
RG-Reference Value: 2,508
captcha-rule Value: 1,213
Sources
app/mendoza/r Value: 3,590
app/brown/o Value: 3,482
HTTP Methods
GET Value: 3,912
OPTIONS Value: 3,683
DELETE Value: 3,388
PUT Value: 3,317
POST Value: 3,127
Country or Region By Blocked Requests
Top 5
Top WebACLs
Web ACL ID
Web ACL Name
Count
captcha-web-acl 8.00 K
hello-world 3.47 K
sql-injection-acl 3.46 K
rg-reference-acl 2.51 K
Top Sources
Http Source Id
Count
app/mendoza/r 3.59 K
3.50 K
app/brown/o 3.48 K
3.47 K
app/rice/z 3.39 K
Top Request URIs
httpRequest.uri
Count
category 1.18 K
posts 985
blog/search 747
app 736
explore 735
Top Countries or Regions
Country or Region
Count
MN 4.20 K
EG 4.14 K
MV 3.62 K
MW 2.74 K
BY 2.73 K
Top Rules
Terminating Rule Id
Terminating Rule Type
Count
Default_Action REGULAR 13.7 K
RG-Reference GROUP 2.51 K
captcha-rule REGULAR 1.21 K
Top Client IPs
httpRequest.clientIp
Count
4.80.100.203 221
207.136.180.202 215
213.137.96.245 212
89.224.226.120 211
24.92.143.19 209
Top Blocked / Allowed Host URI
uri
Action
Count
category ALLOW 914
posts ALLOW 771
blog/search ALLOW 595
explore ALLOW 588
app ALLOW 575
Top Labels with Host, URI
Label
httpRequest.uri
Count
category 988
posts 864
blog/search 643
explore 634
main 627
Metrics
1 - 15 of 10425 rows

VPC Flow Logs

VPC Flow Logs enable you to capture information about the IP traffic going to and from network interfaces in your VPC.

You can create a log ingestion into Amazon OpenSearch Service either by using the Centralized Logging with OpenSearch console or by deploying a standalone CloudFormation stack.

> **Important**
>
> - Centralized Logging with OpenSearch supports VPCs who publish the flow log data to an Amazon S3 bucket or a CloudWatch log group. When publishing to S3, the S3 bucket region must be the same as the Centralized Logging with OpenSearch solution region.
> - The Amazon OpenSearch Service index is rotated on a daily basis by default, and you can adjust the index in the Additional Settings.

Create log ingestion (Amazon OpenSearch Service for log analytics)

Using the Centralized Logging with OpenSearch Console

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the navigation pane, under **Log Analytics Pipelines**, choose **Service Log**.
3. Choose the **Create a log ingestion** button.
4. In the **AWS Services** section, choose **VPC Flow Logs**.
5. Choose **Next**.
6. Under **Specify settings**, choose **Automatic** or **Manual** for **VPC Flow Log enabling**. The automatic mode will enable the VPC Flow Log and save the logs to a centralized S3 bucket if logging is not enabled yet.
   - For **Automatic mode**, choose the VPC from the dropdown list.
   - For **Manual mode**, enter the **VPC Name** and **VPC Flow Logs location**.
   - (Optional) If you are ingesting VPC Flow logs from another account, select a linked account from the **Account** dropdown list first.
7. Under **Log Source**, select **S3** or **CloudWatch** as the source.
8. Choose **Next**.
9. In the **Specify OpenSearch domain** section, select an imported domain for **Amazon OpenSearch Service domain**.

10. Choose **Yes** for **Sample dashboard** if you want to ingest an associated built-in Amazon OpenSearch Service dashboard.
11. You can change the **Index Prefix** of the target Amazon OpenSearch Service index if needed. The default prefix is your VPC name.
12. In the **Log Lifecycle** section, enter the number of days to manage the Amazon OpenSearch Service index lifecycle. The Centralized Logging with OpenSearch will create the associated Index State Management (ISM) policy automatically for this pipeline.
13. In the **Select log processor** section, choose the log processor.
    - When selecting Lambda as log processor, you can configure the Lambda concurrency if needed.
    - (Optional) OSI as log processor is now supported in these Regions. When OSI is selected, enter the minimum and maximum number of OCU. For more information, see Scaling pipelines.
14. Choose **Next**.
15. Add tags if needed.
16. Choose **Create**.

Using the standalone CloudFormation Stack

This automated AWS CloudFormation template deploys the *Centralized Logging with OpenSearch - VPC Flow Logs Ingestion* solution in the AWS Cloud.

	Launch in AWS Management Console	**Download Template**
AWS Regions	Launch Stack	Template
AWS China Regions	Launch Stack	Template

1. Log in to the AWS Management Console and select above button to launch the AWS CloudFormation template. You can also download the template as a starting point for your own implementation.
2. To launch the stack in a different AWS Region, use the Region selector in the console navigation bar.

3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following parameters.

Parameter	Default	Description
Log Bucket Name	*Requires input*	The S3 bucket name which stores the logs.
Log Bucket Prefix	*Requires input*	The S3 bucket path prefix which stores the logs.
Log Source Account ID	<Optional>	The AWS Account ID of the S3 bucket. Required for cross-account log ingestion (Please add a member account first). By default, the Account ID you logged in at **Step 1** will be used.
Log Source Region	<Optional>	The AWS Region of the S3 bucket. By default, the Region you selected at **Step 2** will be used.
Log Source Account Assume Role	<Optional>	The IAM Role ARN used for cross-account log ingestion . Required for cross-account log ingestion (Please add a member account first).
Engine Type	OpenSearch	The engine type of the OpenSearch. Select OpenSearch.

Parameter	Default	Description
OpenSearch Domain Name	*Requires input*	The domain name of the Amazon OpenSearch Service cluster.
OpenSearch Endpoint	*Requires input*	The OpenSearch endpoint URL. For example, vpc-your_opensearch_domain_name-xcvgw6uu2o6za fsiefxubwuohe.us-east-1.es.amazonaws.com
Index Prefix	*Requires input*	The common prefix of OpenSearch index for the log. The index name will be <Index Prefix>-<Log Type>-<Other Suffix>.
Create Sample Dashboard	Yes	Whether to create a sample OpenSearch dashboard.
VPC ID	*Requires input*	Select a VPC which has access to the OpenSearch domain. The log processor Lambda function will reside in the selected VPC.
Subnet IDs	*Requires input*	Select at least two subnets which have access to the OpenSearch domain. The log processor Lambda function will reside in the subnets. Make sure the subnets have access to the Amazon S3 service.

Parameter	Default	Description
Security Group ID	*Requires input*	Select a Security Group which will be associated with the log processor Lambda function. Make sure the Security Group has access to the OpenSearch domain.
S3 Backup Bucket	*Requires input*	The S3 backup bucket name to store the failed ingestion logs.
KMS-CMK ARN	<Optional>	The KMS-CMK ARN for encryption. Leave it blank to create a new KMS CMK.
Number Of Shards	5	Number of shards to distribute the index evenly across all data nodes. Keep the size of each shard between 10-50 GB.
Number of Replicas	1	Number of replicas for OpenSearch Index. Each replica is a full copy of an index.
Age to Warm Storage	<Optional>	The age required to move the index into warm storage (e.g. 7d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). This is only effective when warm storage is enabled in OpenSearch.

Parameter	Default	Description
Age to Cold Storage	<Optional>	The age required to move the index into cold storage (e.g. 30d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). This is only effective when cold storage is enabled in OpenSearch.
Age to Retain	<Optional>	The age to retain the index (e.g. 180d). Index age is the time between its creation and the present. Supported units are d (days) and h (hours). If value is "", the index will not be deleted.
Rollover Index Size	<Optional>	The minimum size of the shard storage required to roll over the index (e.g. 30GB).
Index Suffix	yyyy-MM-dd	The common suffix format of OpenSearch index for the log(Example: yyyy-MM-dd, yyyy-MM-dd-HH). The index name will be <Index Prefix>-<Log Type>-<Index Suffix>-000001.
Compression type	best_compression	The compression type to use to compress stored data. Available values are best_compression and default.

Parameter	Default	Description
Refresh Interval	1s	How often the index should refresh, which publishes its most recent changes and makes them available for searching. Can be set to -1 to disable refreshing. Default is 1s.
EnableS3Notification	True	An option to enable or disable notifications for Amazon S3 buckets. The default option is recommend ed for most cases.
LogProcessorRoleName	<Optional>	Specify a role name for the log processor. The name should NOT duplicate an existing role name. If no name is specified, a random name is generated.
QueueName	<Optional>	Specify a queue name for an SQS. The name should NOT duplicate an existing queue name. If no name is given, a random name is generated.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template creates AWS Identity and Access Management (IAM) resources.
9. Choose **Create** stack to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately 10 minutes.

View dashboard

The dashboard includes the following visualizations.

Visualization Name	Source Field	Description
Global Filters	• account-id • region • vpc-id • subnet-id • action • flow-direction • log-status • protocol-code • type	The charts are filtered according to Account ID, Region, VPC ID and other conditions.
Total Requests	• log event	Shows the total number of network requests logged by VPC Flow Logs during a selected time period.
Request History	• log event	Presents a bar chart that displays the distribution of events over time.
Requests By VPC ID	• vpc-id	Displays the proportional breakdown of network requests by source VPC using a pie chart.
Total Requests By Action	• action	Displays the total volume of requests segmented by action over time.
Total Bytes	• bytes	Provides visibility into overall bandwidth usage and traffic patterns across the monitored VPCs, subnets, network

Visualization Name	Source Field	Description
		interfaces and security groups.
Total Packets	• packets	Displays total logged packets over time to visualize trends, surges and dips.
Bytes Metric	• bytes • flow-direction	Shows the distribution of incoming (Ingress) and outgoing (Egress) network traffic volumes in bytes across the range of flows logged by VPC Flow Logs over a time period.
Requests By Direction	• flow-direction	Provides visibility into the proportional composition of incoming versus outgoing requests.
Requests By Direction	• flow-direction	Displays the total number of network flows logged by VPC Flow Logs segmented by traffic direction - Ingress vs Egress.
Requests By Type	• type	Shows the volume of flows for each type. This provides visibility into the protocol composition of network requests traversing the environment.

Visualization Name	Source Field	Description
Top Source Bytes	• srcaddr • bytes	Displays the source IP addresses transmitting the highest outbound volume of data during the selected time period.
Top Destination Bytes	• dstaddr • bytes	Enables you to monitor and analyze outbound traffic from your VPC to external destinati ons.
Top Source Requests	• srcaddr	Allows you to see which resources inside your VPC are initiating external requests.
Top Destination Requests	• dstaddr	Allows you to see which external hosts are being contacted most by your VPC resources.
Requests by Protocol	• protocol-code	Displays network flows logged by VPC Flow Logs segmented by traffic type - TCP, UDP, ICMP etc.
Requests by Status	• log-status	Provides a breakdown of network flows by their traffic status - Accepted, Rejected or Other.
Top Sources AWS Services	• pkt-src-aws-service	Show the proportional distribution of flows originati ng from top AWS sources like S3, CloudFront, Lambda, etc. during the selected time period.

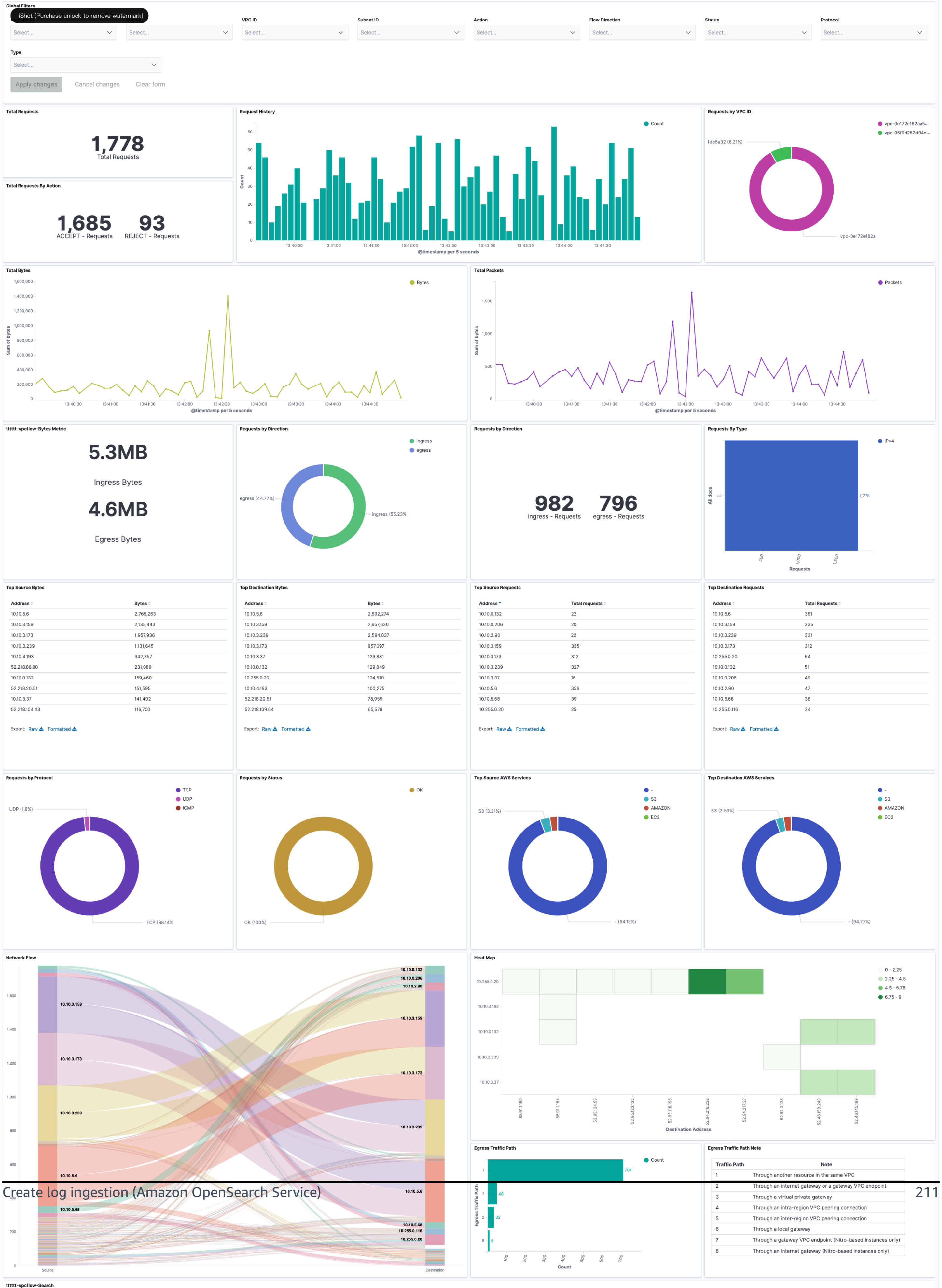
Visualization Name	Source Field	Description
Top Destination AWS Services	• pkt-dst-aws-service	Provide visibility into IP traffic going to and from AWS services located outside your VPC. By enabling flow logs on VPC subnets/interfaces and filtering on traffic with an ACCEPT action, you can view outbound flows from your VPC to various AWS services.
Network Flow	• srcaddr • dstaddr	Allows you to view informati on about the IP traffic going to and from network interfaces in your VPC.
Heat Map	• srcaddr • dstaddr	Offers a visual summary of connections between source and destination IPs in your flow log data.
Egress Traffic Path	• traffic-path	Allows you to enable flow logging on VPC network interfaces to capture information about all IP traffic going to and from that interface.

Visualization Name	Source Field	Description
Search	• @timestamp • account-id • vpc-id • flow-direction • action • protocol-code • srcaddr • scaport • dstaddr • dstport • bytes • packets • log-status	Searching through the detailed flow log data allows pinpoint analysis of traffic around security events, network issues, changes in usage patterns, and more.

Sample Dashboard

You can access the built-in dashboard in Amazon OpenSearch Service to view log data. For more information, see Access Dashboard.

You can click the below image to view the high-resolution sample dashboard.


Global Filters
VPC ID
Subnet ID
Action
Flow Direction
Status
Protocol
Type
Apply changes
Cancel changes
Clear form
Total Requests
1,778
Total Requests
Total Requests By Action
1,685
ACCEPT - Requests
93
REJECT - Requests
Request History
Count
@timestamp per 5 seconds
Requests by VPC ID
Total Bytes
Bytes
Sum of bytes
Total Packets
Packets
ttttt-vpcflow-Bytes Metric
5.3MB
Ingress Bytes
4.6MB
Egress Bytes
Requests by Direction
ingress
egress
egress (44.77%)
ingress (55.23%)
Requests by Direction
982
ingress - Requests
796
egress - Requests
Requests By Type
IPv4
1,778
Requests
Top Source Bytes
Address Bytes
10.10.5.6 2,765,263
10.10.3.159 2,135,443
10.10.3.173 1,957,936
10.10.3.239 1,131,645
10.10.4.193 342,357
52.218.88.80 231,089
10.10.0.132 159,460
52.218.20.51 151,595
10.10.3.37 141,492
52.218.104.43 116,700
Export: Raw Formatted
Top Destination Bytes
Address Bytes
10.10.5.6 2,692,274
10.10.3.159 2,657,630
10.10.3.239 2,594,837
10.10.3.173 957,097
10.10.3.37 129,881
10.10.0.132 129,849
10.255.0.20 124,510
10.10.4.193 100,275
52.218.20.51 78,959
52.218.109.64 65,579
Export: Raw Formatted
Top Source Requests
Address Total requests
10.10.0.132 22
10.10.0.206 20
10.10.2.90 22
10.10.3.159 335
10.10.3.173 312
10.10.3.239 327
10.10.3.37 16
10.10.5.6 356
10.10.5.68 39
10.255.0.20 25
Export: Raw Formatted
Top Destination Requests
Address Total Requests
10.10.5.6 361
10.10.3.159 335
10.10.3.239 331
10.10.3.173 312
10.255.0.20 64
10.10.0.132 51
10.10.0.206 49
10.10.2.90 47
10.10.5.68 38
10.255.0.116 34
Export: Raw Formatted
Requests by Protocol
TCP
UDP
ICMP
UDP (1.8%)
TCP (98.14%)
Requests by Status
OK
OK (100%)
Top Source AWS Services
S3
AMAZON
EC2
S3 (3.21%)
- (94.15%)
Top Destination AWS Services
S3
AMAZON
EC2
S3 (2.59%)
- (94.77%)
Network Flow
Source
Destination
Heat Map
0 - 2.25
2.25 - 4.5
4.5 - 6.75
6.75 - 9
Destination Address
Egress Traffic Path
Count
707
48
32
9
Egress Traffic Path Note
Traffic Path Note
1 Through another resource in the same VPC
2 Through an internet gateway or a gateway VPC endpoint
3 Through a virtual private gateway
4 Through an intra-region VPC peering connection
5 Through an inter-region VPC peering connection
6 Through a local gateway
7 Through a gateway VPC endpoint (Nitro-based instances only)
8 Through an internet gateway (Nitro-based instances only)
ttttt-vpcflow-Search

Create log ingestion (Light Engine for log analytics)

Using the Centralized Logging with OpenSearch Console

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the navigation pane, under **Log Analytics Pipelines**, choose **Service Log**.
3. Choose the **Create a log ingestion** button.
4. In the **AWS Services** section, choose **Amazon VPC Flow**.
5. Choose **Light Engine**, and choose **Next**.
6. Under **Specify settings**, choose **Automatic** or **Manual** for **VPC Flow logs enabling**. The automatic mode will detect the VPC Flow log location automatically.
   - For **Automatic mode**, choose the VPC Flow from the dropdown list.
   - For Standard Log, the solution will automatically detect the log location if logging is enabled.
   - For **Manual mode**, enter the **VPC Flow ID** and **VPC Flow Log location**.
   - (Optional) If you are ingesting CloudFront logs from another account, select a linked account from the **Account** dropdown list first.
7. Choose **Next**.
8. In the **Specify Light Engine Configuration** section, if you want to ingest an associated templated Grafana dashboard, select **Yes** for the sample dashboard.
9. Choose an existing Grafana, or import a new one by making configurations in Grafana.
10. Select an Amazon S3 bucket to store partitioned logs and give a name to the log table. The solution provides a predefined table name, but you can modify it according to your needs.
11. Modify the log processing frequency if needed, which is set to **5** minutes by default with a minimum processing frequency of **1** minute.
12. In the **Log Lifecycle** section, if needed, enter the log merge time and log archive time to modify the default values provided by the solution.
13. Choose **Next**.
14. Add tags if needed.
15. Choose **Create**.

Using the CloudFormation Stack

This automated AWS CloudFormation template deploys the *Centralized Logging with OpenSearch - VpcFlow Standard Log Ingestion* template in the AWS Cloud.

	Launch in AWS Management Console	Download Template
AWS Regions	Launch Stack	Template
AWS China Regions	Launch Stack	Template

1. Log in to the AWS Management Console and select the button to launch the AWS CloudFormation template. You can also download the template as a starting point for your own implementation.
2. To launch the stack in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following parameters.
   - Parameters for **Pipeline settings**

Parameter	Default	Description
Pipeline Id	*Requires input*	The unique identifier for the pipeline, which is essential if you need to create multiple ALB pipelines and write different ALB logs into separate tables. To ensure uniqueness, you can generate a unique pipeline identifier using uuidgenerator.

Parameter	Default	Description
Staging Bucket Prefix	AWSLogs/VpcFlowLogs	The storage directory for logs in the temporary storage area should ensure uniqueness and non-overlapping of the prefix for different pipelines.

- Parameters for **Destination settings**

Parameter	Default	Description
Centralized Bucket Name	*`Requires input`*	The name for the centralized S3 bucket. For example, `centralized-logging-bucket`.
Centralized Bucket Prefix	datalake	The centralized bucket prefix. By default, the database location is `s3://{Centralized Bucket Name}/{Centralized Bucket Prefix}/amazon_cl_centralized`.
Centralized Table Name	VpcFlow	Table name for writing data to the centralized database. You can modify it if needed.

- Parameters for **Scheduler settings**

Parameter	Default	Description
LogProcessor Schedule Expression	rate(5 minutes)	Task scheduling expression for performing log processin g, with a default value of executing the LogProcessor every 5 minutes. For more information, see Schedule types.
LogMerger Schedule Expression	cron(0 1 * * ? *)	Task scheduling expression for performing log merging, with a default value of executing the LogMerger at 1 AM every day. For more information, see Schedule types.
LogArchive Schedule Expression	cron(0 2 * * ? *)	Task scheduling expression for performing log archiving , with a default value of executing the LogArchive at 2 AM every day. For more information, see Schedule types.
Age to Merge	7	Small file retention days, with a default value of 7, indicating that logs older than 7 days will be merged into small files. It can be adjusted as needed.

Parameter	Default	Description
Age to Archive	30	Log retention days, with a default value of 30, indicating that data older than 30 days will be archived and deleted. It can be adjusted as needed.

- Parameters for **Notification settings**

Parameter	Default	Description
Notification Service	SNS	Notification method for alerts. If your main stack is in AWS China Regions, you can only choose the SNS method. If your main stack is in AWS Regions, you can choose either the SNS or SES method.

Parameter	Default	Description
Recipients	*`Requires input`*	If the Notification Service is SNS, enter the SNS Topic ARN to ensure that you have the required permissions. If the Notification Service is SES, enter the email addresses separated by commas to ensure that the email addresses are already Verified Identities in SES. The adminEmail provided during the creation of the main stack will receive a verification email by default.

- Parameters for **Dashboard settings**

Parameter	Default	Description
Import Dashboards	FALSE	Whether to import the Dashboard into Grafana. By default, it is `false`. If it is set to `true`, you must provide the Grafana URL and Grafana Service Account Token.
Grafana URL	*`Requires input`*	Grafana access URL. For example, `https://alb-72277319.us-west-2.elb.amazonaws.com` .

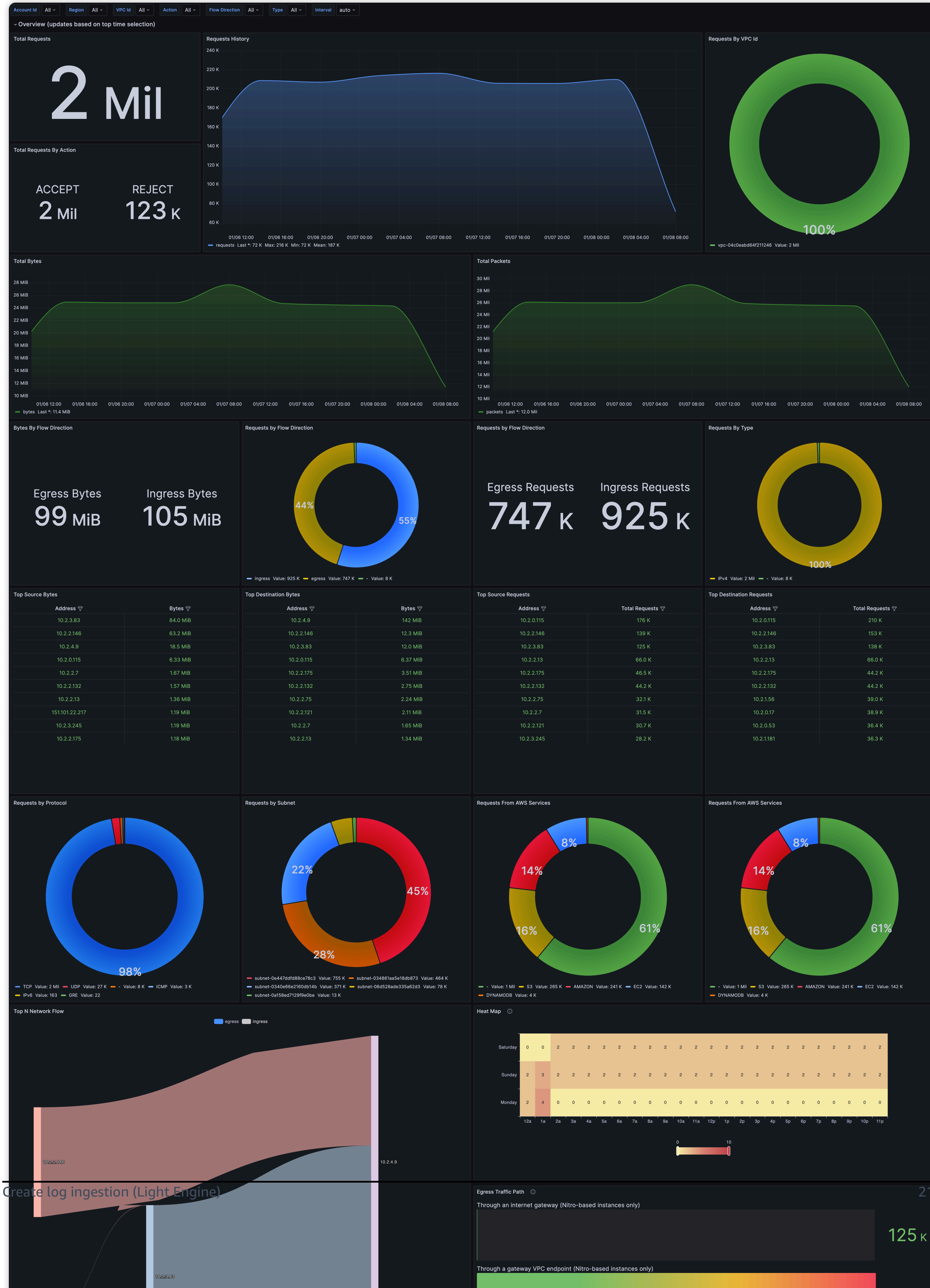
Parameter	Default	Description
Grafana Service Account Token	*Requires input*	Service Account Token created in Grafana.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template creates AWS Identity and Access Management (IAM) resources.
9. Choose **Create** stack to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately 10 minutes.

Sample Dashboard

Below shows the sample dashboard.


Account Id All Region All VPC Id All Action All Flow Direction All Type All Interval auto
Overview (updates based on top time selection)
Total Requests
2 Mil
Requests History
Requests By VPC Id
100%
Total Requests By Action
ACCEPT
2 Mil
REJECT
123 K
Total Bytes
Total Packets
Bytes By Flow Direction
Egress Bytes
99 MiB
Ingress Bytes
105 MiB
Requests by Flow Direction
44%
55%
Requests by Flow Direction
Egress Requests
747 K
Ingress Requests
925 K
Requests By Type
100%
Top Source Bytes
Address Bytes
10.2.3.83 84.0 MiB
10.2.2.146 63.2 MiB
10.2.4.9 18.5 MiB
10.2.0.115 6.33 MiB
10.2.2.7 1.67 MiB
10.2.2.132 1.57 MiB
10.2.2.13 1.36 MiB
151.101.22.217 1.19 MiB
10.2.3.245 1.19 MiB
10.2.2.175 1.18 MiB
Top Destination Bytes
Address Bytes
10.2.4.9 142 MiB
10.2.2.146 12.3 MiB
10.2.3.83 12.0 MiB
10.2.0.115 6.37 MiB
10.2.2.175 3.51 MiB
10.2.2.132 2.75 MiB
10.2.2.75 2.24 MiB
10.2.2.121 2.11 MiB
10.2.2.7 1.65 MiB
10.2.2.13 1.34 MiB
Top Source Requests
Address Total Requests
10.2.0.115 176 K
10.2.2.146 139 K
10.2.3.83 125 K
10.2.2.13 66.0 K
10.2.2.175 46.5 K
10.2.2.132 44.2 K
10.2.2.75 32.1 K
10.2.2.7 31.5 K
10.2.2.121 30.7 K
10.2.3.245 28.2 K
Top Destination Requests
Address Total Requests
10.2.0.115 210 K
10.2.2.146 153 K
10.2.3.83 138 K
10.2.2.13 66.0 K
10.2.2.175 44.2 K
10.2.2.132 44.2 K
10.2.1.56 39.0 K
10.2.0.17 38.9 K
10.2.0.53 36.4 K
10.2.1.181 36.3 K
Requests by Protocol
98%
Requests by Subnet
22%
45%
28%
Requests From AWS Services
8%
14%
16%
61%
Requests From AWS Services
8%
14%
16%
61%
Top N Network Flow
egress ingress
Heat Map
Saturday
Sunday
Monday
Egress Traffic Path
Through an internet gateway (Nitro-based instances only)
125 K
Through a gateway VPC endpoint (Nitro-based instances only)

AWS Config Logs

By default, AWS Config delivers configuration history and snapshot files to your Amazon S3 bucket.

Create log ingestion

You can create a log ingestion into Amazon OpenSearch Service either by using the Centralized Logging with OpenSearch console or by deploying a standalone CloudFormation stack.

Important

- AWS Config must be enabled in the same region as the Centralized Logging with OpenSearch solution.
- The Amazon OpenSearch Service index is rotated on a daily basis by default, and you can adjust the index in the Additional Settings.

Using the Centralized Logging with OpenSearch Console

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the navigation pane, under **Log Analytics Pipelines**, choose **Service Log**.
3. Choose the **Create a log ingestion** button.
4. In the **AWS Services** section, choose **AWS Config Logs**.
5. Choose **Next**.
6. Under **Specify settings**, choose **Automatic** or **Manual** for **Log creation**.
   - For **Automatic mode**, make sure the S3 bucket location is correct, and enter the **AWS Config Name**.
   - For **Manual mode**, enter the **AWS Config Name** and **Log location**.
   - (Optional) If you are ingesting AWS Config logs from another account, select a linked account from the **Account** dropdown list first.
7. Choose **Next**.
8. In the **Specify OpenSearch domain** section, select an imported domain for **Amazon OpenSearch Service domain**.
9. Choose **Yes** for **Sample dashboard** if you want to ingest an associated built-in Amazon OpenSearch Service dashboard.

10. You can change the **Index Prefix** of the target Amazon OpenSearch Service index if needed. The default prefix the AWS Config Name you entered in previous steps.
11. In the **Log Lifecycle** section, enter the number of days to manage the Amazon OpenSearch Service index lifecycle. The Centralized Logging with OpenSearch will create the associated Index State Management (ISM) policy automatically for this pipeline.
12. Choose **Next**.
13. Add tags if needed.
14. Choose **Create**.

Using the standalone CloudFormation Stack

This automated AWS CloudFormation template deploys the *Centralized Logging with OpenSearch - AWS Config Log Ingestion* solution in the AWS Cloud.

	Launch in AWS Management Console	Download Template
AWS Regions	Launch Stack	Template
AWS China Regions	Launch Stack	Template

1. Log in to the AWS Management Console and select above button to launch the AWS CloudFormation template. You can also download the template as a starting point for your own implementation.
2. To launch the stack in a different AWS Region, use the Region selector in the console navigation bar.
3. On the **Create stack** page, verify that the correct template URL shows in the **Amazon S3 URL** text box and choose **Next**.
4. On the **Specify stack details** page, assign a name to your solution stack.
5. Under **Parameters**, review the parameters for the template and modify them as necessary. This solution uses the following default values.

Parameter	Default	Description
Log Bucket Name	*Requires input*	The S3 bucket name which stores the logs.
Log Bucket Prefix	*Requires input*	The S3 bucket path prefix which stores the logs.
Log Source Account ID	<Optional>	The AWS Account ID of the S3 bucket. Required for cross-account log ingestion (Please [link an account](#) first). By default, the Account ID you logged in at **Step 1** will be used.
Log Source Region	<Optional>	The AWS Region of the S3 bucket. By default, the Region you selected at **Step 2** will be used.
Log Source Account Assume Role	<Optional>	The IAM Role ARN used for cross-account log ingestion . Required for cross-account log ingestion (Please [link an account](#) first).
Engine Type	OpenSearch	The engine type of the OpenSearch. Select OpenSearch.
OpenSearch Domain Name	*Requires input*	The domain name of the Amazon OpenSearch Service cluster.

Parameter	Default	Description
OpenSearch Endpoint	*Requires input*	The OpenSearch endpoint URL. For example, vpc-your_opensearch_domain_name-xcvgw6uu2o6zafsiefxubwuohe.us-east-1.es.amazonaws.com
Index Prefix	*Requires input*	The common prefix of OpenSearch index for the log. The index name will be <Index Prefix>-<log-type>-<Index Suffix>-<00000x>.
Create Sample Dashboard	Yes	Whether to create a sample OpenSearch dashboard.
VPC ID	*Requires input*	Select a VPC which has access to the OpenSearch domain. The log processor Lambda function will reside in the selected VPC.
Subnet IDs	*Requires input*	Select at least two subnets which have access to the OpenSearch domain. The log processor Lambda function will reside in the subnets. Make sure the subnets have access to the Amazon S3 service.

Parameter	Default	Description
Security Group ID	*Requires input*	Select a Security Group which will be associated with the log processor Lambdafun ction . Make sure the Security Group has access to the OpenSearch domain.
S3 Backup Bucket	*Requires input*	The S3 backup bucket name to store the failed ingestion logs.
KMS-CMK ARN	<Optional>	The KMS-CMK ARN for SQS encryption. Leave it blank to create a new KMS CMK.
Number Of Shards	5	Number of shards to distribute the index evenly across all data nodes. Keep the size of each shard between 10-50 GiB.
Number of Replicas	1	Number of replicas for OpenSearch Index. Each replica is a full copy of an index.
Age to Warm Storage	<Optional>	The age required to move the index into warm storage (for example, 7 days). Index age is the time elapsed from its creation until now. Supported units are d (days) and h (hours). This takes effect only when warm storage is enabled in OpenSearch.

Parameter	Default	Description
Age to Cold Storage	<Optional>	The age required to move the index into cold storage (for example, 30 days). Index age is the time elapsed from its creation until now. Supported units are d (days) and h (hours). This takes effect only when cold storage is enabled in OpenSearch.
Age to Retain	<Optional>	The age to retain the index (for example, 180 days). Index age is the time elapsed from its creation until now. Supported units are d (days) and h (hours). If value is "", the index will not be deleted.
Rollover Index Size	<Optional>	The minimum size of the shard storage required to roll over the index (for example, 30GB).
Index Suffix	YYYY-MM-DD	The common suffix format of OpenSearch index for the log (for example, YYYY-MM-DD, YYYY-MM-DD-HH).
Compression type	best_compression	The compression type used to compress stored data. Available values are best_compression and default.

Parameter	Default	Description
Refresh Interval	1s	How often the index will be refreshed to publish its most recent changes and make them available for searching. You can set it to -1 to disable refreshing. Default is 1s.

6. Choose **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review** page, review and confirm the settings. Check the box acknowledging that the template creates AWS Identity and Access Management (IAM) resources.
9. Choose **Create** stack to deploy the stack.

You can view the status of the stack in the AWS CloudFormation console in the **Status** column. You should receive a **CREATE_COMPLETE** status in approximately 10 minutes.

View dashboard

The dashboard includes the following visualizations.

Visualization Name	Source Field	Description
Global Filters	• awsAccountId • awsRegion • resourceType • resourceId • resourceName	The charts are filtered according to Account ID, Region, VPC ID and other conditions.
Total Change Events	• log event	Shows the number of configuration changes detected across all AWS resources during a selected time period.

Visualization Name	Source Field	Description
Top Resource Types	• resourceType	Displays the breakdown of configuration changes by the most frequently modified AWS resource types during a selected time period.
Config History	• log event	Presents a bar chart that displays the distribution of events over time.
Total Delete Events	• log event	Shows the number of AWS resource deletion events detected by AWS Config during a selected time period.
Config Status	• configurationItemStatus	Displays the operational state of the AWS Config service across monitored regions and accounts.
Top S3 Changes	• resourceName	Displays the Amazon S3 buckets undergoing the highest number of configura tion changes during a selected time period.
Top Changed Resources	• resourceName • resourceId • resourceType	Displays the individual AWS resources undergoin g the highest number of configuration changes during a selected time period.

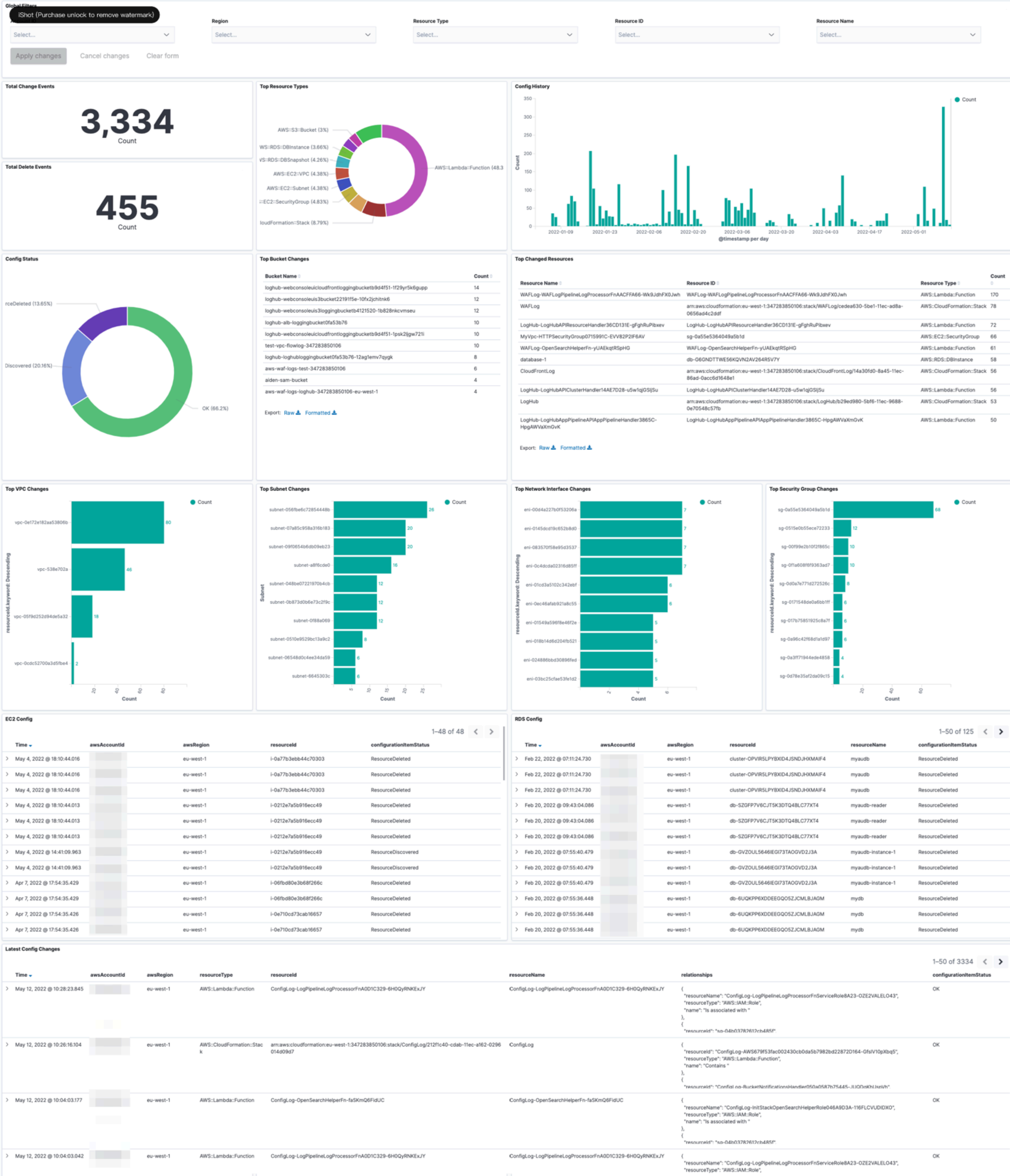
Visualization Name	Source Field	Description
Top VPC Changes	• resourceId	Presents a bar chart that Displays the Amazon VPCs undergoing the highest number of configuration changes during a selected time period.
Top Subnet Changes	• resourceId	Delivers targeted visibility into the subnets undergoin g the most transformation for governance, security and stability.
Top Network Interface Changes	• resourceId	Spotlights the Amazon VPC network interfaces seeing the most configuration changes during a selected period.
Top Security Group Changes	• resourceId	Top 10 changed groups rank by total modification count.
EC2 Config	• @timestamp • awsAccountId • resourceId • configurationItemStatus	Allows reconstructing the incremental changes applied to EC2 configurations over time for auditing.
RDS Config	• @timestamp • awsAccountId • awsRegion • resourceId • resourceName • configurationItemStatus	Shows the configuration history and changes detected by AWS Config for RDS database resources

Visualization Name	Source Field	Description
Latest Config Changes	• @timestamp • awsAccountId • awsRegion • resourceType • resourceId • resourceName • relationships • configurationItemStatus	Offers an at-a-glance overview of infrastructure modifications.

Sample Dashboard

You can access the built-in dashboard in Amazon OpenSearch Service to view log data. For more information, see Access Dashboard.

You can click the below image to view the high-resolution sample dashboard.



Application Logs

Centralized Logging with OpenSearch supports ingesting application logs from the following log sources.

- Amazon EC2 instance group: the solution automatically installs log agent Fluent Bit 1.9), collects application logs on EC2 instances and then sends logs into Amazon OpenSearch Service.
- Amazon EKS cluster: the solution generates all-in-one configuration file for customers to deploy the log agent (Fluent Bit 1.9) as a DaemonSet or Sidecar. After log agent is deployed, the solution starts collecting pod logs and sends them to Amazon OpenSearch Service.
- Amazon S3: the solution either ingests logs in the specified Amazon S3 location continuously or performs one-time ingestion. You can also filter logs based on Amazon S3 prefix or parse logs with custom Log Config.
- Syslog: the solution collects syslog logs through UP or TCP protocol.

Amazon OpenSearch Service is suitable for real-time log analytics and frequent queries and has full-text search capability.

As of release 2.1.0, the solution starts to support log ingestion into Light Engine, which is suitable for non real-time log analytics and infrequent queries and has SQL-like search capability. The feature is supported when you choose Amazon EC2 instance group or Amazon EKS cluster as log source.

After creating a log analytics pipeline, you can add more log sources to the log analytics pipeline. For more information, see add a new log source.

> **Important**
>
> If you are using the Centralized Logging with OpenSearch to create an application log pipeline for the first time, you are recommended to learn the concepts and the supported log formats and log sources.

Supported Log Formats and Log Sources

The table lists the log formats supported by each log source. For more information about how to create log ingestion for each log format, refer to Log Config.

Log Format	Amazon EC2 Instance Group	Amazon EKS Cluster	Amazon S3	Syslog
Nginx	Yes	Yes	Yes	No
Apache HTTP Server	Yes	Yes	Yes	No
JSON	Yes	Yes	Yes	Yes
Single-line Text	Yes	Yes	Yes	Yes
Multi-line Text	Yes	Yes	Yes	No
Multi-line Text (Spring Boot)	Yes	Yes	Yes	No
Syslog RFC5424/RFC3164	No	No	No	Yes
Syslog Custom	No	No	No	Yes

Concepts

The following introduce concepts that help you to understand how the application log ingestion works.

Application Log Analytics Pipeline

To collect application logs, a data pipeline is needed. The pipeline not only buffers the data in transmit but also cleans or pre-processes data. For example, transforming IP to Geo location. Currently, Kinesis Data Stream is used as data buffering for EC2 log source.

Log Ingestion

A log ingestion configures the Log Source, Log Config and the Application Log Analytics Pipeline for the log agent used by Centralized Logging with OpenSearch. After that, Centralized Logging with OpenSearch will start collecting certain type of logs from the log source and send them to Amazon OpenSearch Service.

Log Agent

A log agent is a program that reads logs from one location and sends them to another location (for example, OpenSearch). Currently, Centralized Logging with OpenSearch only supports Fluent Bit 1.9 log agent which is installed automatically. The Fluent Bit agent has a dependency of OpenSSL 1.1. To learn how to install OpenSSL on Linux instances, refer to OpenSSL installation. To find the supported platforms by Fluent Bit, refer to this link.

Log Buffer

Log Buffer is a buffer layer between the Log Agent and OpenSearch clusters. The agent uploads logs into the buffer layer before being processed and delivered into the OpenSearch clusters. A buffer layer is a way to protect OpenSearch clusters from overwhelming. This solution provides the following types of buffer layers.

- **Amazon S3**. Use this option if you can bear minutes-level latency for log ingestion. The log agent periodically uploads logs to an Amazon S3 bucket. The frequency of data delivery to Amazon S3 is determined by *Buffer size* (default value is 50 MiB) and *Buffer interval* (default value is 60 seconds) value that you configured when creating the application log analytics pipelines. The condition satisfied first triggers data delivery to Amazon S3.
- **Amazon Kinesis Data Streams**. Use this option if you need real-time log ingestion. The log agent uploads logs to Amazon Kinesis Data Stream in seconds. The frequency of data delivery to Kinesis Data Streams is determined by *Buffer size* (10 MiB) and *Buffer interval* (5 seconds). The condition satisfied first triggers data delivery to Kinesis Data Streams.

Log Buffer is optional when creating an application log analytics pipeline. For all types of application logs, this solution allows you to ingest logs without any buffer layers. However, we only recommend this option when you have small log volume, and you are confident that the logs will not exceed the thresholds at the OpenSearch side.

Log Source

A Log Source refers to a location where you want Centralized Logging with OpenSearch to collect application logs from. Supported log sources includes:

- Amazon EC2 Instance Group
- Amazon EKS Cluster
- Amazon S3
- Syslog

Amazon EC2 Instance Group

An instance group is a collection of EC2 instances from which you want to collect application logs. Centralized Logging with OpenSearch can help you install the log agent in each instance within a group. You can select arbitrary instances through the user interface, or choose an EC2 Auto Scaling group.

Amazon EKS Cluster

The EKS Cluster in Centralized Logging with OpenSearch refers to the Amazon EKS from which you want to collect pod logs. Centralized Logging with OpenSearch will guide you to deploy the log agent as a DaemonSet or Sidecar in the EKS Cluster.

Amazon S3

Centralized Logging with OpenSearch supports collectings logs stored in an Amazon S3 bucket.

Syslog

Centralized Logging with OpenSearch supports collecting syslog logs through UDP or TCP protocol.

Log Config

A Log Config is a configuration that defines the format of logs (that is, what fields each log line includes, and the data type of each field), based on which the Log Analytics Pipeline parses the logs before ingesting them into log storage. Log Config also allows you to define filters of the logs based on the fields in the logs.

Amazon EC2 instance group as log source

An instance group represents a group of EC2 Linux instances, which enables the solution to associate a Log Config with multiple EC2 instances quickly. Centralized Logging with OpenSearch uses Systems Manager Agent (SSM Agent) to install/configure Fluent Bit agent, and sends log data to Kinesis Data Streams.

The following guides you to create a log pipeline that ingests logs from an Amazon EC2 instance group.

Create a log analytics pipeline (Amazon OpenSearch Service)

Prerequisites

Make sure you have imported an Amazon OpenSearch Service domain. For more information, see Domain operations.

Follow the steps below:

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the left sidebar, under **Log Analytics Pipelines**, choose **Application Log**.
3. Choose **Create a pipeline**.
4. Choose **Instance Group** as Log Source, choose **Amazon OpenSearch Service**, and choose **Next**.
5. Select an instance group. If you have no instance group yet, choose **Create Instance Group** at the top right corner, and follow the instructions to create an instance group. After that, choose **Refresh** and then select the newly created instance group.
6. (Auto Scaling group only) If your instance group is created based on an Auto Scaling group, after ingestion status become "Created", then you can find the generated Shell Script in the instance group's detail page. Copy the shell script and update the User Data of the Auto Scaling Launch configurations or Launch template.
7. Keep the default **Permission grant method**.
8. (Optional) If you choose **I will manually add the below required permissions after pipeline creation**, continue to do the following:
   a. Choose **Expand to view required permissions** and copy the provided JSON policy.
   b. Go to AWS Management Console.
   c. On the left navigation pane, choose **IAM**, and select **Policies** under **Access management**.
   d. Choose **Create Policy**, choose **JSON** and replace all the content inside the text block. Make sure to substitute <YOUR ACCOUNT ID> with your account id.
   e. Choose **Next**, and then enter a name for this policy.

f. Attach the policy to your EC2 instance profile to grant the log agent permissions to send logs to the application log pipeline. If you are using Auto Scaling group, you need to update the IAM instance profile associated with the Auto Scaling group. If needed, you can follow the documentation to update your launch template or launch configuration.

9. Choose **Next**.

You have created a log source for the log analytics pipeline. Now you are ready to make further configurations for the log analytics pipeline with Amazon EC2 instance group as log source.

1. Select a log config. If you do not find the desired log config from the drop-down list, choose **Create New**, and follow instructions in Log Config.
2. Enter a **Log Path** to specify the location of logs to be collected.
3. Specify **Index name** in lowercase.
4. In the **Buffer** section, choose **S3** or **Kinesis Data Streams**. If you don't want the buffer layer, choose **None**. Refer to the Log Buffer for more information about choosing the appropriate buffer layer.
   - S3 buffer parameters

Parameter	Default	Description
S3 Bucket	*A log bucket will be created by the solution.*	You can also select a bucket to store the log data.
S3 Bucket Prefix	`AppLogs/<index-prefix>/year=%Y/month=%m/day=%d`	The log agent appends the prefix when delivering the log files to the S3 bucket.
Buffer size	50 MiB	The maximum size of log data cached at the log agent side before delivering to S3. For more information, see Data Delivery Frequency.
Buffer interval	60 seconds	The maximum interval of the log agent to deliver logs to S3. For more information,

Parameter	Default	Description
		see Data Delivery Frequency .
Compression for data records	Gzip	The log agent compresses records before delivering them to the S3 bucket.

- Kinesis Data Streams buffer parameters

Parameter	Default	Description
Shard number	*Requires input*	The number of shards of the Kinesis Data Streams. Each shard can have up to 1,000 records per second and total data write rate of 1MB per second.
Enable auto scaling	No	This solution monitors the utilization of Kinesis Data Streams every 5 minutes, and scale in/out the number of shards automatically. The solution will scale in/out for a maximum of 8 times within 24 hours.
Maximum Shard number	*Requires input*	Required if auto scaling is enabled. The maximum number of shards.

> **Important**
>
> You may observe duplicate logs in OpenSearch if threshold error occurs in Kinesis Data Streams (KDS). This is because the Fluent Bit log agent uploads logs in chunk (contains multiple records), and will retry the chunk if upload failed. Each KDS shard can support

up to 1,000 records per second for writes, up to a maximum total data write rate of 1 MB per second. Please estimate your log volume and choose an appropriate shard number.

5. Choose **Next**.
6. In the **Specify OpenSearch domain** section, select an imported domain for **Amazon OpenSearch Service domain**.
7. In the **Log Lifecycle** section, enter the number of days to manage the Amazon OpenSearch Service index lifecycle. The Centralized Logging with OpenSearch will create the associated Index State Management (ISM) policy automatically for this pipeline.
8. In the **Select log processor** section, choose the log processor.
   - When selecting Lambda as log processor, you can configure the Lambda concurrency if needed.
   - (Optional) OSI as log processor is now supported in these Regions. When OSI is selected, enter the minimum and maximum number of OCU. For more information, see Scaling pipelines.
9. Choose **Next**.
10. Enable **Alarms** if needed and select an exiting SNS topic. If you choose **Create a new SNS topic**, please provide a name and an email address for the new SNS topic.
11. Add tags if needed.
12. Choose **Create**.
13. Wait for the application pipeline to turn to "Active" state.

Create a log analytics pipeline (Light Engine)

Follow the steps below:

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the left sidebar, under **Log Analytics Pipelines**, choose **Application Log**.
3. Choose **Create a pipeline**.
4. Choose **Instance Group** as Log Source, choose **Light Engine**, and choose **Next**.
5. Select an instance group. If you have no instance group yet, choose **Create Instance Group** at the top right corner, and follow the instructions to create an instance group. After that, choose **Refresh** and then select the newly created instance group.

6. (Auto Scaling group only) If your instance group is created based on an Auto Scaling group, after ingestion status become "Created", then you can find the generated Shell Script in the instance group's detail page. Copy the shell script and update the User Data of the Auto Scaling Launch configurations or Launch template.
7. Keep the default **Permission grant method**.
8. (Optional) If you choose **I will manually add the below required permissions after pipeline creation**, continue to do the following:
   a. Choose **Expand to view required permissions** and copy the provided JSON policy.
   b. Go to AWS Management Console.
   c. On the left navigation pane, choose **IAM**, and select **Policies** under **Access management**.
   d. Choose **Create Policy**, choose **JSON** and replace all the content inside the text block. Make sure to substitute <YOUR ACCOUNT ID> with your account id.
   e. Choose **Next**, and then enter a name for this policy.
   f. Attach the policy to your EC2 instance profile to grant the log agent permissions to send logs to the application log pipeline. If you are using Auto Scaling group, you need to update the IAM instance profile associated with the Auto Scaling group. If needed, you can follow the documentation to update your launch template or launch configuration.
9. Choose **Next**.

You have created a log source for the log analytics pipeline. Now you are ready to make further configurations for the log analytics pipeline with Amazon EC2 instance group as log source.

1. Select a log config. If you do not find the desired log config from the drop-down list, choose **Create New**, and follow instructions in Log Config.
2. Enter a **Log Path** to specify the location of logs to be collected.
3. In the **Buffer** section, configure Amazon S3 buffer parameters.

Parameter	Default	Description
S3 Bucket	*A log bucket will be created by the solution.*	You can also select a bucket to store the log data.
Buffer size	50 MiB	The maximum size of log data cached at the log agent

Parameter	Default	Description
		side before delivering to S3. For more information, see [Data Delivery Frequency]().
Buffer interval	60 seconds	The maximum interval of the log agent to deliver logs to S3. For more information, see [Data Delivery Frequency]().
Compression for data records	Gzip	The log agent compresses records before delivering them to the S3 bucket.

4. Choose **Next**.
5. In the **Specify Light Engine Configuration** section, if you want to ingest an associated templated Grafana dashboard, select **Yes** for the sample dashboard.
6. Choose an existing Grafana, or import a new one by making configurations in Grafana.
7. Select an Amazon S3 bucket to store partitioned logs and give a name to the log table. The solution provides a predefined table name, but you can modify it according to your needs.
8. Modify the log processing frequency if needed, which is set to **5** minutes by default with a minimum processing frequency of **1** minute.
9. In the **Log Lifecycle** section, enter the log merger time and lag archive time. The solution provides default values, which you can modify according to your needs.
10. Choose **Next**.
11. Enable **Alarms** if needed and select an exiting SNS topic. If you choose **Create a new SNS topic**, please provide a name and an email address for the new SNS topic.
12. Add tags if needed.
13. Choose **Create**.
14. Wait for the application pipeline to turn to "Active" state.

Amazon EKS cluster as log source

For Amazon Elastic Kubernetes Service (Amazon EKS) clusters, Centralized Logging with OpenSearch generates an all-in-one configuration file for you to deploy the log agent (Fluent Bit 1.9) as a DaemonSet or Sidecar. After log agent is deployed, the solution starts collecting pod logs and send them to Amazon OpenSearch Service.

The following guides you to create a log pipeline that ingests logs from an Amazon EKS cluster.

Create a log analytics pipeline (Amazon OpenSearch Service)

Prerequisites

Make sure you have imported an Amazon OpenSearch Service domain. For more information, see Domain operations.

Follow the steps below:

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the left sidebar, under **Log Analytics Pipelines**, choose **Application Log**.
3. Choose **Create a pipeline**.
4. Choose **Amazon EKS** as Log Source, and choose **Next**.
5. Choose the AWS account in which the logs are stored.
6. Choose an EKS Cluster. If no clusters are imported yet, choose **Import an EKS Cluster** and follow instructions to import an EKS cluster. After that, select the newly imported EKS cluster from the drop-down list.
7. Choose **Next**.

You have created a log source for the log analytics pipeline. Now you are ready to make further configurations for the log analytics pipeline with Amazon EKS cluster as log source.

1. Select a log config. If you do not find the desired log config from the drop-down list, choose **Create New** and follow instructions in Log Config.
2. Enter a **Log Path** to specify the location of logs you want to collect.
3. Specify **Index name** in lowercase.

4. In the **Buffer** section, choose **S3** or **Kinesis Data Streams**. If you don't want the buffer layer, choose **None**. Refer to the Log Buffer for more information about choosing the appropriate buffer layer.

   - S3 buffer parameters

Parameter	Default	Description
S3 Bucket	*A log bucket will be created by the solution.*	You can also select a bucket to store the log data.
S3 Bucket Prefix	AppLogs/<index-prefix>/year=%Y/month=%m/day=%d	The log agent appends the prefix when delivering the log files to the S3 bucket.
Buffer size	50 MiB	The maximum size of log data cached at the log agent side before delivering to S3. For more information, see Data Delivery Frequency.
Buffer interval	60 seconds	The maximum interval of the log agent to deliver logs to S3. For more information, see Data Delivery Frequency .
Compression for data records	Gzip	The log agent compresse s records before delivering them to the S3 bucket.

   - Kinesis Data Streams buffer parameters

Parameter	Default	Description
Shard number	*Requires input*	The number of shards of the Kinesis Data Streams. Each shard can have up to 1,000 records per second and total data write rate of 1MB per second.
Enable auto scaling	No	This solution monitors the utilization of Kinesis Data Streams every 5 minutes, and scale in/out the number of shards automatically. The solution will scale in/out for a maximum of 8 times within 24 hours.
Maximum Shard number	*Requires input*	Required if auto scaling is enabled. The maximum number of shards.

Important

You may observe duplicate logs in OpenSearch if threshold error occurs in Kinesis Data Streams (KDS). This is because the Fluent Bit log agent uploads logs in chunk (contains multiple records), and will retry the chunk if upload failed. Each KDS shard can support up to 1,000 records per second for writes, up to a maximum total data write rate of 1 MB per second. Please estimate your log volume and choose an appropriate shard number.

5. Choose **Next**.
6. In the **Specify OpenSearch domain** section, select an imported domain for **Amazon OpenSearch Service domain**.

7. In the **Log Lifecycle** section, enter the number of days to manage the Amazon OpenSearch Service index lifecycle. The Centralized Logging with OpenSearch will create the associated Index State Management (ISM) policy automatically for this pipeline.
8. In the **Select log processor** section, choose the log processor.
   - When selecting Lambda as log processor, you can configure the Lambda concurrency if needed.
   - (Optional) OSI as log processor is now supported in these Regions. When OSI is selected, enter the minimum and maximum number of OCU. For more information, see Scaling pipelines.
9. Choose **Next**.
10. Enable **Alarms** if needed and select an exiting SNS topic. If you choose **Create a new SNS topic**, please provide a name and an email address for the new SNS topic.
11. Add tags if needed.
12. Choose **Create**.
13. Wait for the application pipeline turning to "Active" state.

Create a log analytics pipeline (Light Engine)

Follow the steps below:

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the left sidebar, under **Log Analytics Pipelines**, choose **Application Log**.
3. Choose **Create a pipeline**.
4. Choose **Amazon EKS** as Log Source, choose **Light Engine** and choose **Next**.
5. Choose the AWS account in which the logs are stored.
6. Choose an EKS Cluster. If no clusters are imported yet, choose **Import an EKS Cluster** and follow instructions to import an EKS cluster. After that, select the newly imported EKS cluster from the drop-down list.
7. Choose **Next**.

You have created a log source for the log analytics pipeline. Now you are ready to make further configurations for the log analytics pipeline with Amazon EKS cluster as log source.

1. Select a log config. If you do not find the desired log config from the drop-down list, choose **Create New** and follow instructions in Log Config.

2. Enter a **Log Path** to specify the location of logs you want to collect.
3. In the **Buffer** section, configure Amazon S3 buffer parameters.

   S3 buffer parameters

Parameter	Default	Description
S3 Bucket	*A log bucket will be created by the solution.*	You can also select a bucket to store the log data.
Buffer size	50 MiB	The maximum size of log data cached at the log agent side before delivering to S3. For more information, see Data Delivery Frequency.
Buffer interval	60 seconds	The maximum interval of the log agent to deliver logs to S3. For more information, see Data Delivery Frequency.
Compression for data records	Gzip	The log agent compresse s records before delivering them to the S3 bucket.

4. Choose **Next**.
5. In the **Specify Light Engine Configuration** section, if you want to ingest an associated templated Grafana dashboard, select **Yes** for the sample dashboard.
6. Choose an existing Grafana, or import a new one by making configurations in Grafana.
7. Select an Amazon S3 bucket to store partitioned logs and give a name to the log table. The solution provides a predefined table name, but you can modify it according to your needs.
8. Modify the log processing frequency if needed, which is set to **5** minutes by default with a minimum processing frequency of **1** minute.
9. In the **Log Lifecycle** section, enter the log merger time and lag archive time. The solution provides default values, which you can modify according to your needs.
10. Choose **Next**.

11. Enable **Alarms** if needed and select an exiting SNS topic. If you choose **Create a new SNS topic**, please provide a name and an email address for the new SNS topic.
12. Add tags if needed.
13. Choose **Create**.
14. Wait for the application pipeline turning to "Active" state.

Amazon S3 as log source

For Amazon S3, Centralized Logging with OpenSearch ingests logs in a specified Amazon S3 location continuously or performs one-time ingestion. You can also filter logs based on Amazon S3 prefix or parse logs with custom Log Config.

The following guides you to create a log pipeline that ingests logs from an Amazon S3 bucket.

Prerequisites

Make sure you have imported an Amazon OpenSearch Service domain. For more information, see Domain operations.

Create a log analytics pipeline

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the left sidebar, under **Log Analytics Pipelines**, choose **Application Log**.
3. Choose **Create a pipeline**.
4. Choose **Amazon S3** as Log Source, and choose **Next**.
5. Choose the Amazon S3 bucket where your logs are stored. If needed, enter **Prefix filter**, which is optional.
6. Choose **Ingestion mode** based on your need. If you want to ingest logs continuously, select **On-going**; if you only need to ingest logs once, select **One-time**.
7. Specify **Compression format** if your log files are compressed, and choose **Next**.

You have created a log source for the log analytics pipeline. Now you are ready to make further configurations for the log analytics pipeline with Amazon S3 as log source.

1. Select a log config. If you do not find the desired log config from the drop-down list, choose **Create New**. Refer to Log Config for more information.

2. Choose **Next**.
3. Specify **Index name** in lowercase.
4. In the **Specify OpenSearch domain** section, select an imported domain for **Amazon OpenSearch Service domain**.
5. In the **Log Lifecycle** section, enter the number of days to manage the Amazon OpenSearch Service index lifecycle. The Centralized Logging with OpenSearch creates the associated Index State Management (ISM) policy automatically for this pipeline.
6. Choose **Next**.
7. Enable **Alarms** if needed and select an existing SNS topic. If you choose **Create a new SNS topic**, provide a name and an email address for the new SNS topic.
8. Add tags if needed.
9. Choose **Create**.
10. Wait for the application pipeline to turn to an "Active" state.

Syslog as log source

Centralized Logging with OpenSearch collects syslog logs through UDP or TCP protocol.

The following guides you to create a log pipeline that ingests logs from a syslog endpoint.

Prerequisites

Make sure you have imported an Amazon OpenSearch Service domain. For more information, see Domain operations.

Create a log analytics pipeline

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the left sidebar, under **Log Analytics Pipelines**, choose **Application Log**.
3. Choose **Create a pipeline**.
4. Choose **Syslog Endpoint** as Log Source, and choose **Next**.
5. Select **UDP** or **TCP** with custom port number. Choose **Next**.

You have created a log source for the log analytics pipeline. Now you are ready to make further configurations for the log analytics pipeline with syslog as log source.

1. Select a log config. If you do not find the desired log config from the drop-down list, choose **Create New**. Refer to Log Config for more information.
2. Choose **Next**.
3. Specify **Index name** in lowercase.
4. In the **Buffer** section, choose **S3** or **Kinesis Data Streams**. If you don't want the buffer layer, choose **None**. Refer to the Log Buffer for more information about choosing the appropriate buffer layer.
   - S3 buffer parameters

Parameter	Default	Description
S3 Bucket	*A log bucket will be created by the solution.*	You can also select a bucket to store the log data.
S3 Bucket Prefix	AppLogs/<index-prefix>/year=%Y/month=%m/day=%d	The log agent appends the prefix when delivering the log files to the S3 bucket.
Buffer size	50 MiB	The maximum size of log data cached at the log agent side before delivering to S3. For more information, see Data Delivery Frequency.
Buffer interval	60 seconds	The maximum interval of the log agent to deliver logs to S3. For more information, see Data Delivery Frequency.
Compression for data records	Gzip	The log agent compresses records before delivering them to the S3 bucket.

   - Kinesis Data Streams buffer parameters

Parameter	Default	Description
Shard number	*Requires input*	The number of shards of the Kinesis Data Streams. Each shard can have up to 1,000 records per second and total data write rate of 1MB per second.
Enable auto scaling	No	This solution monitors the utilization of Kinesis Data Streams every 5 minutes, and scale in/out the number of shards automatically. The solution will scale in/out for a maximum of 8 times within 24 hours.
Maximum Shard number	*Requires input*	Required if auto scaling is enabled. The maximum number of shards.

> **Important**
>
> You may observe duplicate logs in OpenSearch if threshold error occurs in Kinesis Data Streams (KDS). This is because the Fluent Bit log agent uploads logs in chunk (contains multiple records), and will retry the chunk if upload failed. Each KDS shard can support up to 1,000 records per second for writes, up to a maximum total data write rate of 1 MB per second. Please estimate your log volume and choose an appropriate shard number.

5. Choose **Next**.
6. In the **Specify OpenSearch domain** section, select an imported domain for **Amazon OpenSearch Service domain**.

7. In the **Log Lifecycle** section, enter the number of days to manage the Amazon OpenSearch Service index lifecycle. The Centralized Logging with OpenSearch will create the associated Index State Management (ISM) policy automatically for this pipeline.
8. In the **Select log processor** section, choose the log processor.
    - When selecting Lambda as log processor, you can configure the Lambda concurrency if needed.
    - (Optional) OSI as log processor is now supported in these Regions. When OSI is selected, enter the minimum and maximum number of OCU. For more information, see Scaling pipelines.
9. Choose **Next**.
10. Enable **Alarms** if needed and select an exiting SNS topic. If you choose **Create a new SNS topic**, please provide a name and an email address for the new SNS topic.
11. Add tags if needed.
12. Choose **Create**.
13. Wait for the application pipeline turning to "Active" state.

Pipeline resources

A log analytics pipeline can have more than one log sources.

Log sources

You need to create a log source first before collecting application logs. Centralized Logging with OpenSearch supports the following log sources:

- Amazon EC2 instance group
- Amazon EKS cluster
- Amazon S3
- Syslog

For more information, see concepts.

Amazon EC2 Instance Group

An instance group represents a group of EC2 Linux instances, which enables the solution to associate a Log Config with multiple EC2 instances quickly. Centralized Logging with OpenSearch uses Systems Manager Agent (SSM Agent) to install/configure Fluent Bit agent, and sends log data to Kinesis Data Streams.

Prerequisites

Make sure the instances meet the following requirements:

- SSM agent is installed on instances. Refer to install SSM agent on EC2 instances for Linux for more details.
- The `AmazonSSMManagedInstanceCore` policy is being associated with the instances.
- The OpenSSL 1.1 or later is installed. Refer to OpenSSL Installation for more details.
- The instances have network access to AWS Systems Manager.
- The instances have network access to Amazon Kinesis Data Streams, if you use it as the Log Buffer.
- The instances have network access to Amazon S3, if you use it as the Log Buffer.
- The operating system of the instances are supported by Fluent Bit. Refer to Supported Platform.

(Option 1) Select instances to create an Instance Group

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the left sidebar, under **Log Source**, choose **Instance Group**.
3. Choose **Create an instance group**.
4. In the **Instance Group Settings** section, specify a group name.
5. Select **Instances**. You can use up to 5 tags to filter the instances.
6. Verify that all the selected instances "Pending Status" is **Online**.
7. (Optional) If the selected instances "Pending Status" are empty, click the **Install log agent** button and wait for "Pending Status" to become **Online**.
8. (Optional) If you want to ingest logs from another account, select a linked account in the **Account Settings** section to create an instance group log source from another account.
9. Choose **Create**.

> **Important**
>
> An installation error may occur if you use the Centralized Logging with OpenSearch console to install Fluent Bit agent on Ubuntu instances in **China (Beijing) Region Operated by Sinnet (cn-north-1)** and **China (Ningxia) Region Operated by NWCD (cn-northwest-1)**.

This is because the Fluent Bit assets cannot be downloaded successfully. You need to install the Fluent Bit agent by yourself.

(Option 2) Select an Auto Scaling group to create an Instance Group

When creating an Instance Group with Amazon EC2 Auto Scaling group, the solution will generate a shell script which you should include in the EC2 User Data.

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the left sidebar, under **Log Source**, choose **Instance Group**.
3. Choose **Create an instance group**.
4. In the **Instance Group Settings** section, specify a group name.
5. Select **Auto Scaling groups**.
6. Select the Auto Scaling group from which you want to collect logs.
7. (Optional) If you want to ingest logs from another account, select a linked account in the **Account Settings** section to create an instance group log source from another account.
8. Choose **Create**. After you created a Log Ingestion using the Instance Group, you can find the generated Shell Script in the details page.
9. Copy the shell script and update the User Data of the Auto Scaling group's launch configurations or launch template. The shell script will automatically install Fluent Bit, SSM agent if needed, and download Fluent Bit configurations.
10. Once you have updated the launch configurations or launch template, you need to start an instance refresh to update the instances within the Auto Scaling group. The newly launched instances will ingest logs to the OpenSearch cluster or the Log Buffer layer.

Amazon EKS cluster

The EKS Cluster in Centralized Logging with OpenSearch refers to the Amazon Elastic Kubernetes Service (Amazon EKS) from which you want to collect pod logs. Centralized Logging with OpenSearch will guide you to deploy the log agent as a DaemonSet or Sidecar in the EKS Cluster.

Important

- Centralized Logging with OpenSearch does not support sending logs in one EKS cluster to more than one Amazon OpenSearch Service domain at the same time.

- Make sure your EKS cluster's VPC is connected to Amazon OpenSearch Service cluster's VPC so that logs can be ingested. Refer to VPC Connectivity for more details regarding approaches to connect VPCs.

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the left sidebar, under **Log Source**, choose **EKS Cluster**.
3. Choose **Import a Cluster**.
4. Choose the **EKS Cluster** where Centralized Logging with OpenSearch collects logs from.
5. (Optional) If you want to ingest logs from another account, select a linked account from the **Account** dropdown to import an EKS log source from another account.
6. Select **DaemonSet** or **Sidecar** as log agent's deployment pattern.
7. Choose **Next**.
8. Specify the **Amazon OpenSearch Service** where Centralized Logging with OpenSearch sends the logs to.
9. Follow the guidance to establish a VPC peering connection between EKS's VPC and OpenSearch's VPC.
   - Create and accept VPC peering connections
   - Update your route tables for a VPC peering connection
   - Update your security groups to reference peer VPC groups
10. Choose **Next**.
11. Add tags if needed.
12. Choose **Create**.

Amazon S3

The S3 in Centralized Logging with OpenSearch refers to the Amazon S3 bucket from which you want to collect application logs. You can choose **On-going** or **One-time** to create your ingestion job.

> **Important**
>
> - On-going means that the ingestion job will run when a new file is delivered to the specified S3 location.

- One-time means that the ingestion job will run at creation and only will run once to load all files in the specified location.

Syslog

Important

To ingest logs, make sure your Syslog generator/sender's subnet is connected to Centralized Logging with OpenSearch's **two** private subnets. Refer to VPC Connectivity for more details about how to connect VPCs.

You can use UDP or TCP custom port number to collect syslog in Centralized Logging with OpenSearch. Syslog refers to logs generated by Linux instance, routers or network equipment. For more information, see Syslog in Wikipedia.

Add a new log source

A newly created log analytics pipeline has one log source. You can add more log sources into the log pipeline.

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the left navigation pane, under **Log Analytics Pipelines**, choose **Application Log**.
3. Choose the log pipeline by clicking its **ID**.
4. Choose **Create a source**.
5. Follow the instructions in Amazon EC2 instance group, Amazon EKS cluster, Amazon S3, or Syslog to create a log source according to your need.

Log Config

Centralized Logging with OpenSearch solution supports creating log configs for the following formats:

- JSON
- Apache
- Nginx

- Syslog
- Single-ine text
- Multi-line text

The following describes how to create log config for each log format.

Create a JSON config

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the left sidebar, under **Resources**, choose **Log Config**.
3. Choose **Create a log config**.
4. Specify **Config Name**.
5. Specify **Log Path**. You can use , to separate multiple paths.
6. Choose **JSON** in the log type dropdown list.
7. In the **Sample log parsing** section, paste a sample JSON log and click **Parse log** to verify if the log parsing is successful. The solution supports nested JSON with a maximum nesting depth of X.

   For example:

   ```
   {"host":"81.95.250.9", "user-identifier":"-", "time":"08/Mar/2022:06:28:03 +0000",
    "method": "PATCH", "request": "/clicks-and-mortar/24%2f7", "protocol":"HTTP/2.0",
    "status":502, "bytes":24337, "referer": "https://www.investorturn-key.net/
   functionalities/innovative/integrated"}
   ```

   If your JSON log sample is nested JSON, choose **Parse log** and it displays a list of field type options for each layer. If needed, you can set the corresponding field type for each layer of fields. If you choose **Remove** to delete a field, the field type will be inferred by OpenSearch automatically.

   For example:

   ```
   {"timestamp": "2023-11-06T08:29:55.266Z",
    "correlationId": "566829027325526589",
    "processInfo": {
     "startTime": "2023-11-06T08:29:55.266Z",
     "hostname": "ltvtix0apidev01",
   ```

```
    "domainId": "e6826d97-a60f-45cb-93e1-b4bb5a7add29",
    "groupId": "group-2",
    "groupName": "grp_dev_bba",
    "serviceId": "instance-1",
    "serviceName": "ins_dev_bba",
    "version": "7.7.20210130"
    },
    "transactionSummary": {
        "path": "https://www.leadmission-critical.info/relationships",
        "protocol": "https",
        "protocolSrc": "97",
        "status": "exception",
        "serviceContexts": [
            {
                "service": "NSC_APP-117127_DCTM_Get Documentum Token",
                "monitor": true,
                "client": "Pass Through",
                "org": null,
                "app": null,
                "method": "getTokenUsingPOST",
                "status": "exception",
                "duration": 25270
            }
                ]
        }
    }
```


8. Check if each fields type mapping is correct. You can change the type by selecting the dropdown menu in the second column. For all supported types, see Data Types.

> **Note**
>
> You must specify the datetime of the log using key “time”. If not specified, system time will be added.

> **Note**
>
> For nested JSON, the Time Key must be on the first level.

9. Specify the **Time format**. The format syntax follows strptime. Check this for details.

10(Optional) In the **Filter** section, you add some conditions to filter logs at the log agent side. The solution will ingest logs that match ALL the specified conditions only.

11Select **Create**.

Create an Apache HTTP server log config

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the left sidebar, under **Resources**, choose **Log Config**.
3. Click the **Create a log config** button.
4. Specify **Config Name**.
5. Specify **Log Path**. You can use , to separate multiple paths.
6. Choose **Apache HTTP server** in the log type dropdown menu.
7. In the **Apache Log Format** section, paste your Apache HTTP server log format configuration. It is in the format of /etc/httpd/conf/httpd.conf and starts with LogFormat.

   For example:

   ```
   LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\"" combined
   ```

8. (Optional) In the **Sample log parsing** section, paste a sample Apache HTTP server log to verify if the log parsing is successful.

   For example:

   ```
   127.0.0.1 - - [22/Dec/2021:06:48:57 +0000] "GET /xxx HTTP/1.1" 404 196 "-"
    "curl/7.79.1"
   ```

9. Choose **Create**.

Create an Nginx log config

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the left sidebar, under **Resources**, choose **Log Config**.
3. Click the **Create a log config** button.
4. Specify **Config Name**.
5. Specify **Log Path**. You can use , to separate multiple paths.

6. Choose **Nginx** in the log type dropdown menu.
7. In the **Nginx Log Format** section, paste your Nginx log format configuration. It is in the format of /etc/nginx/nginx.conf and starts with log_format.

   For example:

```
log_format  main  '$remote_addr - $remote_user [$time_local] "$request" '
'$status $body_bytes_sent "$http_referer" '
'"$http_user_agent" "$http_x_forwarded_for"';
```

8. (Optional) In the **Sample log parsing** section, paste a sample Nginx log to verify if the log parsing is successful.

   For example:

```
127.0.0.1 - - [24/Dec/2021:01:27:11 +0000] "GET / HTTP/1.1" 200 3520 "-"
 "curl/7.79.1" "-"
```

9. (Optional) In the **Filter** section, you add some conditions to filter logs at the log agent side. The solution will ingest logs that match ALL the specified conditions only.

10Select **Create**.

Create a Syslog config

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the left sidebar, under **Resources**, choose **Log Config**.
3. Click the **Create a log config** button.
4. Specify **Config Name**.
5. Choose **Syslog** in the log type dropdown menu. Note that Centralized Logging with OpenSearch also supports Syslog with JSON format and single-line text format.

RFC5424

1. Paste a sample RFC5424 log. For example:

```
<35>1 2013-10-11T22:14:15Z client_machine su - - - 'su root' failed for joe on /dev/
pts/2
```

2. Choose **Parse Log**.

3. Check if each fields type mapping is correct. You can change the type by selecting the dropdown menu in the second column. For all supported types, see Data Types.

> **Note**
>
> You must specify the datetime of the log using key “time”. If not specified, system time will be added.

4. Specify the **Time format**. The format syntax follows strptime. Check this manual for details. For example:

```
%Y-%m-%dT%H:%M:%SZ
```

5. (Optional) In the **Filter** section, you add some conditions to filter logs at the log agent side. The solution will ingest logs that match ALL the specified conditions only.

6. Select **Create**.

RFC3164

1. Paste a sample RFC3164 log. For example:

```
<35>Oct 12 22:14:15 client_machine su: 'su root' failed for joe on /dev/pts/2
```

2. Choose **Parse Log**.

3. Check if each fields type mapping is correct. You can change the type by selecting the dropdown menu in the second column. For all supported types, see Data Types.

> **Note**
>
> You must specify the datetime of the log using key “time”. If not specified, system time will be added. Since there is no year in the timestamp of RFC3164, it cannot be displayed as a time histogram in the Discover interface of Amazon OpenSearch Service.

4. Specify the **Time format**. The format syntax follows strptime. Check this for details. For example:

```
%b %m %H:%M:%S
```

5. (Optional) In the **Filter** section, you add some conditions to filter logs at the log agent side. The solution will ingest logs that match ALL the specified conditions only.
6. Select **Create**.

Custom

1. In the **Syslog Format** section, paste your Syslog log format configuration. It is in the format of /etc/rsyslog.conf and starts with template or $template. The format syntax follows [Syslog Message Format](). For example:

```
<%pri%>1 %timestamp:::date-rfc3339% %HOSTNAME% %app-name% %procid% %msgid% %msg%\n
```

2. In the **Sample log parsing** section, paste a sample Nginx log to verify if the log parsing is successful. For example:

```
<35>1 2013-10-11T22:14:15.003Z client_machine su - - 'su root' failed for joe on /
dev/pts/2
```

3. Check if each fields type mapping is correct. Change the type by selecting the dropdown menu in the second column. For all supported types, see [Data Types]().

> **Note**
>
> You must specify the datetime of the log using key “time”. If not specified, system time will be added.

4. Specify the **Time format**. The format syntax follows [strptime](). Check [this manual]() for details.
5. (Optional) In the **Filter** section, you add some conditions to filter logs at the log agent side. The solution will ingest logs that match ALL the specified conditions only.
6. Select **Create**.

Create a single-line text config

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the left sidebar, under **Resources**, choose **Log Config**.
3. Click the **Create a log config** button.
4. Specify **Config Name**.

5. Specify **Log Path**. You can use , to separate multiple paths.
6. Choose **Single-line Text** in the log type dropdown menu.
7. Write the regular expression in Rubular to validate first and enter the value. For example:

```
(?<remote_addr>\S+)\s*-\s*(?<remote_user>\S+)\s*\[(?<time_local>\d+/\S+/\d
+:\d+:\d+:\d+)\s+\S+\]\s*"(?<request_method>\S+)\s+(?<request_uri>\S+)\s+\S
+"\s*(?<status>\S+)\s*(?<body_bytes_sent>\S+)\s*"(?<http_referer>[^"]*)"\s*"(?
<http_user_agent>[^"]*)"\s*"(?<http_x_forwarded_for>[^"]*)".*
```

8. In the **Sample log parsing** section, paste a sample Single-line text log and click **Parse log** to verify if the log parsing is successful. For example:

```
127.0.0.1 - - [24/Dec/2021:01:27:11 +0000] "GET / HTTP/1.1" 200 3520 "-"
 "curl/7.79.1" "-"
```

9. Check if each fields type mapping is correct. Change the type by selecting the dropdown menu in the second column. For all supported types, see Data Types.

> **Note**
>
> You must specify the datetime of the log using key “time”. If not specified, system time will be added.

10. Specify the **Time format**. The format syntax follows strptime. Check this manual for details.
11. (Optional) In the **Filter** section, you add some conditions to filter logs at the log agent side. The solution will ingest logs that match ALL the specified conditions only.
12. Select **Create**.

Create a multi-line text config

1. Sign in to the Centralized Logging with OpenSearch Console.
2. In the left sidebar, under **Resources**, choose **Log Config**.
3. Click the **Create a log config** button.
4. Specify **Config Name**.
5. Specify **Log Path**. You can use , to separate multiple paths.
6. Choose **Multi-line Text** in the log type dropdown menu.

Java - Spring Boot

1. For Java Spring Boot logs, you could provide a simple log format. For example:

```
%d{yyyy-MM-dd HH:mm:ss.SSS} %-5level [%thread] %logger : %msg%n
```

2. Paste a sample multi-line log. For example:

```
2022-02-18 10:32:26.400 ERROR [http-nio-8080-exec-1]
 org.apache.catalina.core.ContainerBase.[Tomcat].[localhost].[/].
[dispatcherServlet] : Servlet.service() for servlet [dispatcherServlet] in context
 with path [] threw exception [Request processing failed; nested exception is
 java.lang.ArithmeticException: / by zero] with root cause
java.lang.ArithmeticException: / by zero
   at com.springexamples.demo.web.LoggerController.logs(LoggerController.java:22)
   at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
   at java.base/jdk.internal.reflect.NativeMethodAccessorImpl.invoke
```

3. Choose **Parse Log**.
4. Check if each fields type mapping is correct. You can change the type by selecting the dropdown menu in the second column. For all supported types, see Data Types.

> **Note**
>
> You must specify the datetime of the log using key “time”. If not specified, system time will be added.

5. Specify the **Time format**. The format syntax follows strptime. Check this for details.
6. (Optional) In the **Filter** section, you add some conditions to filter logs at the log agent side. The solution will ingest logs that match ALL the specified conditions only.
7. Select **Create**.

Custom

1. For other kinds of logs, you could specify the first line regex pattern. For example:

```
(?<time>\d{4}-\d{2}-\d{2}\s*\d{2}:\d{2}:\d{2}.\d{3})\s*(?<message>goroutine\s*\d\s*
\[.+\]:)
```

2. Paste a sample multi-line log. For example:

```
2023-07-12 10:32:26.400 goroutine 1 [chan receive]:
runtime.gopark(0x4739b8, 0xc420024178, 0x46fcd7, 0xc, 0xc420028e17, 0x3)
  /usr/local/go/src/runtime/proc.go:280 +0x12c fp=0xc420053e30 sp=0xc420053e00
 pc=0x42503c
runtime.goparkunlock(0xc420024178, 0x46fcd7, 0xc, 0x1000f010040c217, 0x3)
  /usr/local/go/src/runtime/proc.go:286 +0x5e fp=0xc420053e70 sp=0xc420053e30
 pc=0x42512e
runtime.chanrecv(0xc420024120, 0x0, 0xc420053f01, 0x4512d8)
  /usr/local/go/src/runtime/chan.go:506 +0x304 fp=0xc420053f20 sp=0xc420053e70
 pc=0x4046b4
runtime.chanrecv1(0xc420024120, 0x0)
  /usr/local/go/src/runtime/chan.go:388 +0x2b fp=0xc420053f50 sp=0xc420053f20
 pc=0x40439b
main.main()
  foo.go:9 +0x6f fp=0xc420053f80 sp=0xc420053f50 pc=0x4512ef
runtime.main()
  /usr/local/go/src/runtime/proc.go:185 +0x20d fp=0xc420053fe0 sp=0xc420053f80
 pc=0x424bad
runtime.goexit()
  /usr/local/go/src/runtime/asm_amd64.s:2337 +0x1 fp=0xc420053fe8 sp=0xc420053fe0
 pc=0x44b4d1
```

3. Choose **Parse Log**.
4. Check if each field type mapping is correct. You can change the type by selecting the dropdown menu in the second column. For all supported types, see Data Types.

> **Note**
>
> You must specify the datetime of the log using key “time”. If not specified, system time will be added.

5. (Optional) In the **Filter** section, you add some conditions to filter logs at the log agent side. The solution will ingest logs that match ALL the specified conditions only.
6. Select **Create**.

Cross-Account Ingestion

Centralized Logging with OpenSearch supports ingesting AWS Service logs and Application logs in different AWS accounts within the same region. After deploying Centralized Logging with OpenSearch in one account (main account), you can launch the CloudFormation stack in a different account (member account), and associate the two accounts (main account and member account) to implement cross-account ingestion.

Concepts

- **Main account**: One account in which you deployed the Centralized Logging with OpenSearch console. The OpenSearch cluster(s) must also be in the same account.
- **Member account**: Another account from which you want to ingest AWS Service logs or application logs.

The CloudFormation stack in the member account has the least privileges. Centralized Logging with OpenSearch need to provision some AWS resources in the member account to collect logs, and will assume an IAM role provisioned in the member account to list or create resources.

For more information, refer to the Architecture section.

Add a member account

Step 1. Launch a CloudFormation stack in the member account

1. Sign in to the Centralized Logging with OpenSearch console.
2. In the navigation pane, under **Resources**, choose **Member Accounts**.
3. Choose the **Link an Account** button. It displays the steps to deploy the CloudFormation stack in the member account.

> **Important**
>
> You need to copy the template URL, which will be used later.

4. Go to the CloudFormation console of the member account.

5. Choose the **Create stack** button and choose **With new resources (standard)**.
6. In the **Create stack** page, enter the template URL you have copied in **Amazon S3 URL**.
7. Follow the steps to create the CloudFormation stack and wait until the CloudFormation stack is provisioned.
8. Go to the **Outputs** tab to check the parameters which will be used in **Step 2**.

Step 2. Link a member account

1. Go back to the Centralized Logging with OpenSearch console.
2. (Optional) In the navigation panel, under **Resources**, choose **Member Accounts**.
3. In **Step 2. Link an account**, enter the parameters using the Outputs parameters from **Step 1**.

Parameter	CloudFormation Outputs	Description
Account Name	N/A	Name of the member account.
Account ID	N/A	12-digit AWS account ID.
Cross Account Role ARN	CrossAccountRoleARN	Centralized Logging with OpenSearch will assume this role to operate resources in the member account.
Fluent Bit Agent Installation Document	AgentInstallDocument	Centralized Logging with OpenSearch will use this SSM Document to install Fluent Bit agent on EC2 instances in the member account.
Fluent Bit Agent Configura tion Document	AgentConfigDocument	Centralized Logging with OpenSearch will use this SSM Document to deliver Fluent Bit configuration to EC2 instances.

Parameter	CloudFormation Outputs	Description
Fluent Bit Agent Installation Document for Windows	AgentInstallDocumentForWindows	Fluent Bit Agent Installation Configuration for Windows.
Fluent Bit Agent Configuration Document for Windows	AnentConfigDocumentForWindows	Fluent Bit Agent Configuration Document.
Fluent Bit Status Check Document	AgentStatusCheckDocument	Status detection of Fluent Bit.
Cross Account S3 Bucket	CrossAccountS3Bucket	You can use the Centralized Logging with OpenSearch console to enable some AWS Service logs and output them to Amazon S3. The logs will be stored in this account.
Cross Account Stack ID	CrossAccountStackId	CloudFormation stack ID in the member account.
Cross Account KMS Key	CrossAccountKMSKeyARN	Centralized Logging with OpenSearch will use the Key Management Services (KMS) key to encrypt Simple Queue Service (SQS).

4. Click the **Link** button.

Log pipeline monitoring

Log alarms

Types of log alarms for this solution include log processor alarms, buffer layer alarms, and source alarms (only for application log pipeline). The alarms will be initiated when the defined condition is met.

Log alarm type	Log alarm condition	Description
Log processor alarms	Error invocation # >= 10 for 5 minutes, 1 consecutive time	When the number of log processor Lambda error calls is greater than or equal to 10 within 5 minutes (including 5 minutes), an email alarm is initiated.
Log processor alarms	Failed record # >= 1 for 1 minute, 1 consecutive time	When the number of failed records is greater than or equal to 1 within a 1-minute window, an alarm will be triggered.
Log processor alarms	Average execution duration in last 5 minutes >= 60000 milliseconds	In the last 5 minutes, when the average execution time of log processor Lambda is greater than or equal to 60 seconds, an email alarm is initiated.
Buffer layer alarms	SQS Oldest Message Age >= 30 minutes	When the age of the oldest SQS message is greater than or equal to 30 minutes, it means that the message has not been consumed for

Log alarm type	Log alarm condition	Description
		at least 30 minutes, and an email alarm is initiated.
Source alarms (only for application log pipeline)	Fluent Bit output_retried_rec ord_total >= 100 for last 5 minutes	When the total number of retry records output by Fluent Bit in the past 5 minutes is greater than or equal to 100, an email alarm is initiated.

You can choose to enable log alarms or disable them according to your needs.

Enable log alarms

1. Sign in to the Centralized Logging with OpenSearch console.
2. In the left navigation bar, under **Log Analytics Pipelines**, choose **AWS Service Log** or **Application Log**.
3. Select the log pipeline created and choose **View details**.
4. Select the **Alarm** tab.
5. Switch on **Alarms** if needed and select an existing SNS topic.
6. If you choose **Create a new SNS topic**, you need to provide email address for the newly-created SNS topic to notify.

Disable log alarms

1. Sign in to the Centralized Logging with OpenSearch console.
2. In the left navigation bar, under **Log Analytics Pipelines**, choose **Application Log** or **Service Log**.
3. Select the log pipeline created and choose **View details**.
4. Select the **Alarm** tab.
5. Switch off **Alarms**.

Monitoring

The following types of metrics are available on the Centralized Logging with OpenSearch console: log source metrics, buffer metrics, and log processor metrics.

Log source metrics

Fluent Bit

- `FluentBitOutputProcRecords` - The number of log records that this output instance has successfully sent. This is the total record count of all unique chunks sent by this output. If a record is not successfully sent, it does not count towards this metric.
- `FluentBitOutputProcBytes` - The number of bytes of log records that this output instance has successfully sent. This is the total byte size of all unique chunks sent by this output. If a record is not sent due to an error, then it does not count towards this metric.
- `FluentBitOutputDroppedRecords` - The number of log records that have been dropped by the output. This means they met an unrecoverable error or retries expired for their chunk.
- `FluentBitOutputErrors` - The number of chunks that have faced an error (either unrecoverable or retrievable). This is the number of times a chunk has failed, and does not correspond with the number of error messages you see in the Fluent Bit log output.
- `FluentBitOutputRetriedRecords` - The number of log records that experienced a retry. This is calculated at the chunk level, and the count increases when an entire chunk is marked for retry. An output plugin might or might not perform multiple actions that generate many error messages when uploading a single chunk.
- `FluentBitOutputRetriesFailed` - The number of times that retries expired for a chunk. Each plugin configures a `Retry_Limit` which applies to chunks. Once the `Retry_Limit` has been reached for a chunk, it is discarded and this metric is incremented.
- `FluentBitOutputRetries` - The number of times this output instance requested a retry for a chunk.

Network Load Balancer

- `SyslogNLBActiveFlowCount` - The total number of concurrent flows (or connections) from clients to targets. This metric includes connections in the `SYN_SENT` and `ESTABLISHED` states. TCP connections are not terminated at the load balancer, so a client opening a TCP connection to a target counts as a single flow.

- `SyslogNLBProcessedBytes` - The total number of bytes processed by the load balancer, including TCP/IP headers. This count includes traffic to and from targets, except for health check traffic.

Buffer metrics

Log Buffer is a buffer layer between the Log Agent and OpenSearch clusters. The agent uploads logs into the buffer layer before the logs are processed and delivered into the OpenSearch clusters. A buffer layer can be used to protect OpenSearch clusters from overwhelming.

Kinesis Data Stream

- `KDSIncomingBytes` – The number of bytes successfully put to the Kinesis data stream over the specified time period. This metric includes bytes from `PutRecord` and `PutRecords` operations. `Minimum`, `Maximum`, and `Average` statistics represent the bytes in a single put operation for the stream in the specified time period.
- `KDSIncomingRecords` – The number of records successfully put to the Kinesis data stream over the specified time period. This metric includes record counts from `PutRecord` and `PutRecords` operations. `Minimum`, `Maximum`, and `Average` statistics represent the records in a single put operation for the stream in the specified time period.
- `KDSPutRecordBytes` – The number of bytes put to the Kinesis data stream using the `PutRecord` operation over the specified time period.
- `KDSThrottledRecords` – The number of records rejected due to throttling in a `PutRecords` operation per Kinesis data stream, measured over the specified time period.
- `KDSWriteProvisionedThroughputExceeded` – The number of records rejected due to throttling for the stream over the specified time period. This metric includes throttling from `PutRecord` and `PutRecords` operations. The most commonly used statistic for this metric is `Average`.

When the `Minimum` statistic has a non-zero value, the solution throttles records for the stream during the specified time period.

When the `Maximum` statistic has a value of 0 (zero), the solution does not throttle records for the stream during the specified time period.

SQS

Amazon SQS emits the `NumberOfMessagesDeleted` metric for every successful deletion operation that uses a valid receipt handle, including duplicate deletions.

- `SQSNumberOfMessagesSent` - The number of messages added to a queue.
- `SQSNumberOfMessagesDeleted` - The number of messages deleted from the queue.

The following scenarios might cause the value of the NumberOfMessagesDeleted metric to be higher than expected:

- Calling the `DeleteMessage` action on different receipt handles that belong to the same message: If the message is not processed before the visibility timeout expires, the message becomes available to other consumers that can process it and delete it again, increasing the value of the `NumberOfMessagesDeleted` metric.
- Calling the `DeleteMessage` action on the same receipt handle: If the message is processed and deleted, but you call the `DeleteMessage` action again using the same receipt handle, a success status is returned, increasing the value of the `NumberOfMessagesDeleted` metric.

After a message is received three times (or more) and not processed, the message is moved to the back of the queue and the `ApproximateAgeOfOldestMessage` metric points at the second-oldest message that hasn't been received more than three times. This action occurs even if the queue has a redrive policy.

- `SQSApproximateNumberOfMessagesVisible` - The number of messages available for retrieval from the queue.
- `SQSApproximateAgeOfOldestMessage` - The approximate age of the oldest non-deleted message in the queue.

Because a single poison-pill message (received multiple times but never deleted) can distort this metric, the age of a poison-pill message isn't included in the metric until the poison-pill message is consumed successfully.

When the queue has a redrive policy, the message is moved to a dead-letter queue after the configured **Maximum Receives**. When the message is moved to the dead-letter queue, the `ApproximateAgeOfOldestMessage` metric of the dead-letter queue represents the time when the message was moved to the dead-letter queue (not the original time the message was sent).

Log processor metrics

The log processor Lambda function is responsible for performing final processing on the data and bulk writing it to OpenSearch.

- `TotalLogs` – The total number of log records or events processed by the Lambda function.
- `ExcludedLogs` – The number of log records or events that were excluded from processing, which could be due to filtering or other criteria.
- `LoadedLogs` – The number of log records or events that were successfully processed and loaded into OpenSearch.
- `FailedLogs` – The number of log records or events that failed to be processed or loaded into OpenSearch.
- `ConcurrentExecutions` – The number of function instances that are processing events. If this number reaches your concurrent executions quota for the Region, or the reserved concurrency limit on the function, then Lambda throttles additional invocation requests.
- `Duration` – The amount of time that your function code spends processing an event. The billed duration for an invocation is the value of `Duration` rounded up to the nearest millisecond.
- `Throttles` – The number of invocation requests that are throttled. When all function instances are processing requests and no concurrency is available to scale up, Lambda rejects additional requests with a `TooManyRequestsException` error. Throttled requests and other invocation errors don't count as either Invocations or Errors.
- `Invocations` – The number of times that your function code is invoked, including successful invocations and invocations that result in a function error. Invocations aren't recorded if the invocation request is throttled or otherwise results in an invocation error. The value of Invocations equals the number of requests billed.

Frequently Asked Questions

General

Q: What is Centralized Logging with OpenSearch solution?

Centralized Logging with OpenSearch is an AWS Solution that simplifies the building of log analytics pipelines. It provides to customers, as complementary of Amazon OpenSearch Service, capabilities to ingest and process both application logs and AWS service logs without writing code, and create visualization dashboards from out-of-the-box templates. Centralized Logging with OpenSearch automatically assembles the underlying AWS services, and provides you a web console to manage log analytics pipelines.

Q: What are the supported logs in this solution?

Centralized Logging with OpenSearch supports both AWS service logs and EC2/EKS application logs. Refer to the supported AWS services, and the supported application log formats and sources for more details.

Q: Does Centralized Logging with OpenSearch support ingesting logs from multiple AWS accounts?

Yes. Centralized Logging with OpenSearch supports ingesting AWS service logs and application logs from a different AWS account in the same region. For more information, see Cross-Account Ingestion.

Q: Does Centralized Logging with OpenSearch support ingesting logs from multiple AWS Regions?

Currently, Centralized Logging with OpenSearch does not automate the log ingestion from a different AWS Region. You need to ingest logs from other regions into pipelines provisioned by Centralized Logging with OpenSearch. For AWS services which store the logs in S3 bucket, you can leverage the S3 Cross-Region Replication to copy the logs to the Centralized Logging with OpenSearch deployed region, and import incremental logs using the manual mode by specifying the log location in the S3 bucket. For application logs on EC2 and EKS, you need to set up the networking (for example, Kinesis VPC endpoint, VPC Peering), install agents, and configure the agents to ingest logs to Centralized Logging with OpenSearch pipelines.

Q: What is the license of this solution?

This solution is provided under the Apache-2.0 license. It is a permissive free software license written by the Apache Software Foundation. It allows users to use the software for any purpose, to distribute it, to modify it, and to distribute modified versions of the software under the terms of the license, without concern for royalties.

Q: How can I submit a feature request or bug report?

You can submit feature requests and bug report through the GitHub issues.

Q: How can I use stronger TLS Protocols to secure traffic, namely TLS 1.2 and later?

By default, CloudFront uses the TLSv1 security policy along with a default certificate. Changing the TLS settings for CloudFront depends on the presence of your SSL certificates. If you don't have your own SSL certificates, you cannot alter the TLS settings for CloudFront.

To configure TLS 1.2 and later, you need a custom domain. This setup will enable you to enforce stronger TLS protocols for your traffic.

To learn how to configure a custom domain and enable TLS 1.2 and later for your service, refer to Use a Custom Domain with AWS AppSync, Amazon CloudFront, and Amazon Route 53.

Setup and configuration

Q: Can I deploy Centralized Logging with OpenSearch on AWS in any AWS Region?

Centralized Logging with OpenSearch provides two deployment options: option 1 with Cognito User Pool, and option 2 with OpenID Connect. For option 1, customers can deploy the solution in AWS Regions where Amazon Cognito User Pool, AWS AppSync, Amazon Data Firehose (optional) are available. For option 2, customers can deploy the solution in AWS Regions where AWS AppSync, Amazon Data Firehose (optional) are available. Refer to supported regions for deployment for more information.

Q: What are the prerequisites of deploying this solution?

Centralized Logging with OpenSearch does not provision Amazon OpenSearch Service clusters, and you need to import existing OpenSearch clusters through the web console. The clusters must meet the requirements specified in prerequisites.

Q: Why do I need a domain name with ICP recordal when deploy the solution in AWS China Regions?

The Centralized Logging with OpenSearch console is served via CloudFront distribution which is considered as an Internet information service. According to the local regulations, any Internet information service must bind to a domain name with ICP recordal.

Q: What versions of OpenSearch does the solution work with?

Centralized Logging with OpenSearch supports Amazon OpenSearch Service, with OpenSearch 1.3 or later.

Q: What are the index name rules for OpenSearch created by the Log Analytics Pipeline?

You can change the index name if needed when using the Centralized Logging with OpenSearch console to create a log analytics pipeline.

If the log analytics pipeline is created for service logs, the index name is composed of <index prefix>-<service-type>-<index suffix>-<00000x>, where you can define a name for index prefix and service-type is automatically generated by the solution according to the service type you have chosen. Moreover, you can choose different index suffix types to adjust index rollover time window.

- YYYY-MM-DD-HH: Amazon OpenSearch Service will roll the index by hour.
- YYYY-MM-DD: Amazon OpenSearch Service will roll the index by 24 hours.
- YYYY-MM: Amazon OpenSearch Service will roll the index by 30 days.
- YYYY: Amazon OpenSearch Service will roll the index by 365 days.

It should be noted that in OpenSearch, the time is in UTC 0 time zone.

Regarding the 00000x part, Amazon OpenSearch Service will automatically append a 6-digit suffix to the index name, where the first index rule is 000001, rollover according to the index, and increment backwards, such as 000002, 000003.

If the log analytics pipeline is created for application log, the index name is composed of <index prefix>--<index suffix>-<00000x>. The rules for index prefix and index suffix, 00000x are the same as those for service logs.

Q: What are the index rollover rules for OpenSearch created by the Log Analytics Pipeline?

Index rollover is determined by two factors. One is the index suffix in the index name. If you enable the index rollover by capacity, Amazon OpenSearch Service will roll your index when the index capacity equals or exceeds the specified size, regardless of the rollover time window. Note that if one of these two factors matches, index rollover can be triggered.

For example, we created an application log pipeline on January 1, 2023, deleted the application log pipeline at 9:00 on January 4, 2023, and the index name is nginx-YYYY-MM-DD-<00000x>. At the same time, we enabled the index rollover by capacity and entered 300GB. If the log data volume increases suddenly after creation, it can reach 300GB every hour, and the duration is 2 hours and 10 minutes. After that, it returns to normal, and the daily data volume is 90GB. Then OpenSearch creates three indexes on January 1, the index names are nginx-2023-01-01-000001, nginx-2023-01-01-000002, nginx-2023-01-01-000003, and then creates one every day Indexes respectively: nginx-2023-01-02-000004, nginx-2023-01-03-000005, nginx-2023-01-04-000006.

Q: Can I deploy the solution in an existing VPC?

Yes. You can either launch the solution with a new VPC or launch the solution with an existing VPC. When using an existing VPC, you need to select the VPC and the corresponding subnets. Refer to launch with Cognito User Pool or launch with OpenID Connect for more details.

Q: I did not receive the email containing the temporary password when launching the solution with Cognito User Pool. How can I resend the password?

Your account is managed by the Cognito User Pool. To resend the temporary password, you can find the user pool created by the solution, delete and recreate the user using the same email address. If you still have the same issue, try with another email address.

Q: How can I create more users for this solution?

If you launched the solution with Cognito User Pool, go to the AWS console, find the user pool created by the solution, and you can create more users. If you launched the solution with OpenID Connect (OIDC), you should add more users in the user pool managed by the OIDC provider. Note that all users have the same privileges.

Pricing

Q: How will I be charged and billed for the use of this solution?

The solution is free to use, and you are responsible for the cost of AWS services used while running this solution. You pay only for what you use, and there are no minimum or setup fees. Refer to the Cost section for detailed cost estimation.

Q: Will there be additional cost for cross-account ingestion?

No. The cost will be same as ingesting logs within the same AWS account.

Log Ingestion

Q: What is the log agent used in the Centralized Logging with OpenSearch solution?

Centralized Logging with OpenSearch uses AWS for Fluent Bit, a distribution of Fluent Bit maintained by AWS. The solution uses this distribution to ingest logs from Amazon EC2 and Amazon EKS.

Q: I have already stored the AWS service logs of member accounts in a centralized logging account. How should I create service log ingestion for member accounts?

In this case, you need to deploy the Centralized Logging with OpenSearch solution in the centralized logging account, and ingest AWS service logs using the *Manual* mode from the logging account. Refer to this guide for ingesting Application Load Balancer logs with *Manual* mode. You can do the same with other supported AWS services which output logs to S3.

Q: Why there are some duplicated records in OpenSearch when ingesting logs via Kinesis Data Streams?

This is usually because there is no enough Kinesis Shards to handle the incoming requests. When threshold error occurs in Kinesis, the Fluent Bit agent will retry that chunk. To avoid this issue, you need to estimate your log throughput and set a proper Kinesis shard number. Please refer to the Kinesis Data Streams quotas and limits. Centralized Logging with OpenSearch provides a built-in feature to scale-out and scale-in the Kinesis shards, and it would take a couple of minutes to scale out to the desired number.

Q: How to install log agent on CentOS 7

1. Log in to your CentOS 7 machine and install SSM Agent manually.

```
sudo yum install -y http://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/
linux_amd64/amazon-ssm-agent.rpm
sudo systemctl enable amazon-ssm-agent
sudo systemctl start amazon-ssm-agent
```

2. Go to the **Instance Group** panel of Centralized Logging with OpenSearch console, create **Instance Group**, select the CentOS 7 machine, choose **Install log agent** and wait for its status to be **offline**.
3. Log in to CentOS 7 and install fluent-bit 1.9.3 manually.

```
export RELEASE_URL=${FLUENT_BIT_PACKAGES_URL:-https://packages.fluentbit.io}
export RELEASE_KEY=${FLUENT_BIT_PACKAGES_KEY:-https://packages.fluentbit.io/
fluentbit.key}

sudo rpm --import $RELEASE_KEY
cat << EOF | sudo tee /etc/yum.repos.d/fluent-bit.repo
[fluent-bit]
name = Fluent Bit
baseurl = $RELEASE_URL/centos/VERSION_ARCH_SUBSTR
gpgcheck=1
repo_gpgcheck=1
gpgkey=$RELEASE_KEY
enabled=1
EOF
sudo sed -i 's|VERSION_ARCH_SUBSTR|\$releasever/\$basearch/|g' /etc/yum.repos.d/
fluent-bit.repo
sudo yum install -y fluent-bit-1.9.3-1

# Modify the configuration file
sudo sed -i 's/ExecStart.*/ExecStart=\/opt\/fluent-bit\/bin\/fluent-bit -c \/opt\/
fluent-bit\/etc\/fluent-bit.conf/g' /usr/lib/systemd/system/fluent-bit.service
sudo systemctl daemon-reload
sudo systemctl enable fluent-bit
sudo systemctl start fluent-bit
```

4. Go back to the **Instance Groups** panel of the Centralized Logging with OpenSearch console and wait for the CentOS 7 machine status to be **Online** and proceed to create the instance group.

Q: How can I consume CloudWatch custom logs?

You can use Firehose to subscribe CloudWatch logs and transfer logs into Amazon S3. Firstly, create subscription filters with Amazon Kinesis Data Firehose based on this guide. Next, follow the documentation to learn how to transfer logs to Amazon S3. Then, you can use Centralized Logging with OpenSearch to ingest logs from Amazon S3 to OpenSearch.

Log Visualization

Q. How can I find the built-in dashboards in OpenSearch?

Please refer to the AWS Service Logs and Application Logs to find out if there is a built-in dashboard supported. You also need to turn on the *Sample Dashboard* option when creating a log analytics pipeline. The dashboard will be inserted into the Amazon OpenSearch Service under **Global Tenant**. You can switch to the Global Tenant from the top right coder of the OpenSearch Dashboards.

Troubleshooting

The following help you to fix errors or problems that you might encounter when using Centralized Logging with OpenSearch.

Error: Failed to assume service-linked role arn:x:x:x:/AWSServiceRoleForAppSync

The reason for this error is that the account has never used the AWS AppSync service. You can deploy the solution's CloudFormation template again. AWS has already created the role automatically when you encountered the error.

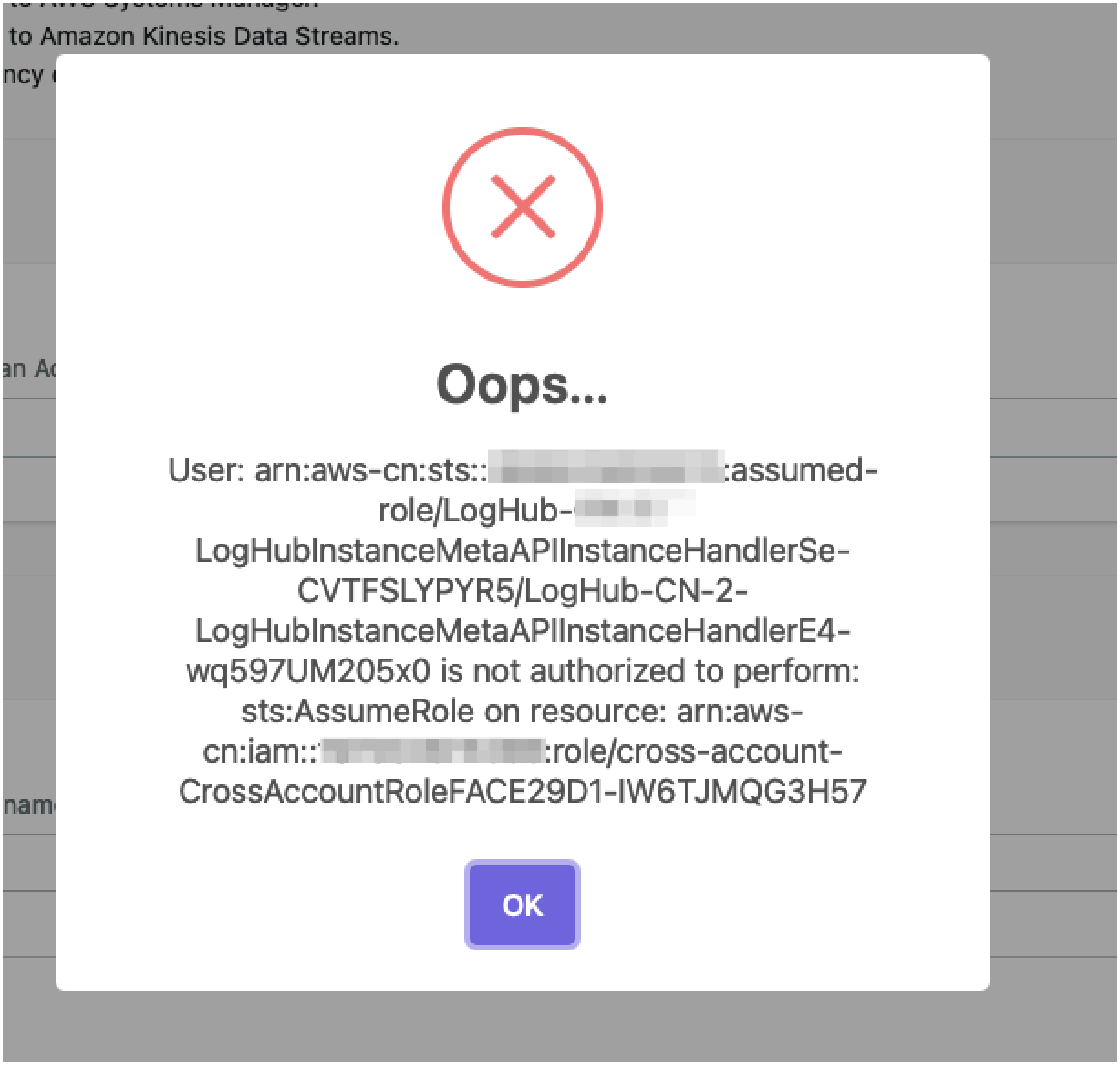
You can also go to AWS CloudShell or the local terminal and run the following AWS CLI command to Link AppSync Role

```
aws iam create-service-linked-role --aws-service-name appsync.amazonaws.com
```

Error: Unable to add backend role

Centralized Logging with OpenSearch only supports Amazon OpenSearch Service domain with Fine-grained access control enabled. You need to go to Amazon OpenSearch Service console, and edit the **Access policy** for the Amazon OpenSearch Service domain.

Error：**User xxx is not authorized to perform sts:AssumeRole on resource**



If you see this error, please make sure you have entered the correct information during cross account setup, and then please wait for several minutes.

Centralized Logging with OpenSearch uses AssumeRole for cross-account access. This is the best practice to temporary access the AWS resources in your sub-account. However, these roles created during cross account setup take seconds or minutes to be affective.

Error: PutRecords API responded with error='InvalidSignatureException'

Fluent-bit agent reports PutRecords API responded with error='InvalidSignatureException', message='The request signature we calculated does not match the signature you provided. Check your AWS Secret Access Key and signing method. Consult the service documentation for details.'

Please restart the fluent-bit agent. For example, on EC2 with Amazon Linux2, run command:

```
sudo service fluent-bit restart
```

Error: PutRecords API responded with error='AccessDeniedException'

Fluent-bit agent deployed on EKS Cluster reports "AccessDeniedException" when sending records to Kinesis. Verify that the IAM role trust relations are correctly set. With the Centralized Logging with OpenSearch console:

1. Open the Centralized Logging with OpenSearch console.
2. In the left sidebar, under **Log Source**, choose **EKS Clusters**.
3. Choose the **EKS Cluster** that you want to check.
4. Click the **IAM Role ARN** which will open the IAM Role in AWS Management Console.
5. Choose the **Trust relationships** to verify that the OIDC Provider, the service account namespace and conditions are correctly set.

You can get more information from Amazon EKS IAM role configuration

My CloudFormation stack is stuck on deleting an AWS::Lambda::Function resource when I update the stack. How to resolve it?

The Lambda function resides in a VPC, and you need to wait for the associated ENI resource to be deleted.

The agent status is offline after I restart the EC2 instance, how can I make it auto start on instance restart?

This usually happens if you have installed the log agent, but restart the instance before you create any Log Ingestion. The log agent will auto restart if there is at least one Log Ingestion. If you have a log ingestion, but the problem still exists, you can use systemctl status fluent-bit to check its status inside the instance.

I have switched to Global tenant. However, I still cannot find the dashboard in OpenSearch.

This is usually because Centralized Logging with OpenSearch received 403 error from OpenSearch when creating the index template and dashboard. This can be fixed by re-run the Lambda function manually by following the steps below:

With the Centralized Logging with OpenSearch console:

1. Open the Centralized Logging with OpenSearch console, and find the AWS Service Log pipeline which has this issue.
2. Copy the first 5 characters from the ID section. E.g. you should copy c169c from ID c169cb23-88f3-4a7e-90d7-4ab4bc18982c
3. Go to AWS Management Console > Lambda. Paste in function filters. This will filter in all the lambda function created for this AWS Service Log ingestion.
4. Click the Lambda function whose name contains "OpenSearchHelperFn".
5. In the **Test** tab, create a new event with any Event name.
6. Click the **Test** button to trigger the Lambda, and wait the lambda function to complete.
7. The dashboard should be available in OpenSearch.

Error from Fluent-bit agent: version `GLIBC_2.25' not found

This error is caused by old version of glibc. Centralized Logging with OpenSearch with version later than 1.2 requires glibc-2.25 or above. So you must upgrade the existing version in EC2 first. The upgrade command for different kinds of OS is shown as follows:

> **Important**
>
> Important We strongly recommend you run the commands with environments first. Any upgrade failure may cause severe loss.

Redhat 7.9

For Redhat 7.9, the whole process will take about 2 hours,and at least 10 GB storage is needed.

```
# install library
yum install -y gcc gcc-c++ m4 python3 bison  fontconfig-devel  libXpm-devel texinfo
 bzip2 wget
echo /usr/local/lib  >> /etc/ld.so.conf

# create tmp directory
mkdir -p /tmp/library
cd /tmp/library

# install gmp-6.1.0
wget https://ftp.gnu.org/gnu/gmp/gmp-6.1.0.tar.bz2
tar xjvf gmp-6.1.0.tar.bz2
cd gmp-6.1.0
./configure --prefix=/usr/local
make && make install
ldconfig
cd ..

# install mpfr-3.1.4
wget https://gcc.gnu.org/pub/gcc/infrastructure/mpfr-3.1.4.tar.bz2
tar xjvf mpfr-3.1.4.tar.bz2
cd mpfr-3.1.4
./configure --with-gmp=/usr/local --prefix=/usr/local
make && make install
ldconfig
cd ..
```

```
# install mpc-1.0.3
wget https://gcc.gnu.org/pub/gcc/infrastructure/mpc-1.0.3.tar.gz
tar xzvf mpc-1.0.3.tar.gz
cd mpc-1.0.3
./configure --prefix=/usr/local
make && make install
ldconfig
cd ..

# install gcc-9.3.0
wget https://ftp.gnu.org/gnu/gcc/gcc-9.3.0/gcc-9.3.0.tar.gz
tar xzvf gcc-9.3.0.tar.gz
cd gcc-9.3.0
mkdir build
cd build/
../configure --enable-checking=release --enable-language=c,c++ --disable-multilib --
prefix=/usr
make -j4 && make install
ldconfig
cd ../..

# install make-4.3
wget https://ftp.gnu.org/gnu/make/make-4.3.tar.gz
tar xzvf make-4.3.tar.gz
cd make-4.3
mkdir build
cd build
../configure --prefix=/usr
make && make install
cd ../..

# install glibc-2.31
wget https://ftp.gnu.org/gnu/glibc/glibc-2.31.tar.gz
tar xzvf glibc-2.31.tar.gz
cd glibc-2.31
mkdir build
cd build/
../configure  --prefix=/usr --disable-profile --enable-add-ons --with-headers=/usr/
include --with-binutils=/usr/bin --disable-sanity-checks --disable-werror
make all && make install
make localedata/install-locales

# clean tmp directory
```

```
cd /tmp
rm -rf /tmp/library
```

Ubuntu 22

```
sudo ln -s /snap/core20/1623/usr/lib/x86_64-linux-gnu/libcrypto.so.1.1 /usr/lib/x86_64-
linux-gnu/libcrypto.so.1.1
sudo ln -s /snap/core20/1623/usr/lib/x86_64-linux-gnu/libssl.so.1.1 /usr/lib/x86_64-
linux-gnu/libssl.so.1.1
sudo ln -s /usr/lib/x86_64-linux-gnu/libsasl2.so.2 /usr/lib/libsasl2.so.3
```

Amazon Linux 2023

```
sudo su -

yum install -y wget perl unzip gcc zlib-devel
mkdir /tmp/openssl
cd /tmp/openssl
wget https://www.openssl.org/source/openssl-1.1.1s.tar.gz
tar xzvf openssl-1.1.1s.tar.gz
cd openssl-1.1.1s
./config --prefix=/usr/local/openssl11 --openssldir=/usr/local/openssl11 shared zlib
make
make install

echo /usr/local/openssl11/lib/ >> /etc/ld.so.conf
ldconfig
```

Uninstall the solution

You will encounter IAM role missing error if you delete the Centralized Logging with OpenSearch main stack before you delete the log pipelines. Centralized Logging with OpenSearch console launches additional CloudFormation stacks to ingest logs. If you want to uninstall the Centralized Logging with OpenSearch solution. We recommend you to delete log pipelines (incl. AWS Service log pipelines and application log pipelines) before uninstall the solution.

Step 1. Delete Application Log Pipelines

Important

Please delete all the log ingestion before deleting an application log pipeline.

1. Go to the Centralized Logging with OpenSearch console, in the left sidebar, choose **Application Log**.
2. Click the application log pipeline to view details.
3. In the ingestion tab, delete all the application log ingestion in the pipeline.
4. Uninstall/Disable the Fluent Bit agent.
   - EC2 (Optional): after removing the log ingestion from EC2 instance group. Fluent Bit will automatically stop ship logs, it is optional for you to stop the Fluent Bit in your instances. Here are the command for stopping Fluent Bit agent.

     ```
     sudo service fluent-bit stop
     sudo systemctl disable fluent-bit.service
     ```
   - EKS DaemonSet (Mandatory): if you have chosen to deploy the Fluent Bit agent using DaemonSet, you need to delete your Fluent Bit agent. Otherwise, the agent will continue ship logs to Centralized Logging with OpenSearch pipelines.

     ```
     kubectl delete -f ~/fluent-bit-logging.yaml
     ```
   - EKS SideCar (Mandatory): please remove the fluent-bit agent in your .yaml file, and restart your pod.
5. Delete the Application Log pipeline.
6. Repeat step 2 to Step 5 to delete all your application log pipelines.

Step 2. Delete AWS Service Log Pipelines

1. Go to the Centralized Logging with OpenSearch console, in the left sidebar, choose **AWS Service Log**.
2. Select and delete the AWS Service Log Pipeline one by one.

Step 3. Clean up imported OpenSearch domains

1. Delete Access Proxy, if you have created the proxy using Centralized Logging with OpenSearch console.
2. Delete Alarms, if you have created alarms using Centralized Logging with OpenSearch console.
3. Delete VPC peering Connection between Centralized Logging with OpenSearch's VPC and OpenSearch's VPC.
   a. Go to AWS VPC Console.
   b. Choose **Peering connections** in left sidebar.
   c. Find and delete the VPC peering connection between the Centralized Logging with OpenSearch's VPC and OpenSearch's VPC. You may not have Peering Connections if you did not use the "Automatic" mode when importing OpenSearch domains.
4. (Optional) Remove imported OpenSearch Domains. (This will not delete the Amazon OpenSearch Service domain in the AWS account.)

Step 4. Delete Centralized Logging with OpenSearch stack

1. Go to the CloudFormation console.
2. Find CloudFormation Stack of the Centralized Logging with OpenSearch solution.
3. (Optional) Delete S3 buckets created by Centralized Logging with OpenSearch.

> **Important**
>
> The S3 bucket whose name contains **LoggingBucket** is the centralized bucket for your AWS service log. You might have enabled AWS Services to send logs to this S3 bucket. Deleting this bucket will cause AWS Services failed to send logs.

a. Choose the CloudFormation stack of the Centralized Logging with OpenSearch solution, and select the **Resources** tab.
b. In search bar, enter AWS::S3::Bucket. This will show all the S3 buckets created by Centralized Logging with OpenSearch solution, and the **Physical ID** field is the S3 bucket name.
c. Go to S3 console, and find the S3 bucket using the bucket name. **Empty** and **Delete** the S3 bucket.

4. Delete the CloudFormation Stack of the Centralized Logging with OpenSearch solution.

Additional resources

Grafana

This section introduces how to set up Grafana environment. If you want the solution to generate dashboards in Grafana automatically, you need to perform the following deployment. If you only want to store the data in Amazon S3 without creating dashboards, you can skip this section.

Step 1: Install Grafana

> **Note**
>
> Skip this step if you already have a Grafana environment.

Prerequisite:

An EC2 instance has been launched, supporting both x86 and ARM architecture.

The following steps provide an example using `m6g.medium` instance type, ARM architecture, and Amazon 2023. For more details, refer to [Install Grafana](#).

```
# Edit/etc/yum.repos.d/grafana.repo file#input below content
[grafana]
name=grafana
baseurl=https://rpm.grafana.com
repo_gpgcheck=1
enabled=1
gpgcheck=1
gpgkey=https://rpm.grafana.com/gpg.key
sslverify=1
sslcacert=/etc/pki/tls/certs/ca-bundle.crt

# install grafana
yum install -y grafana

# Start grafana#and check its running status
systemctl start grafana-server
systemctl status grafana-server
```

```
# grafana listens on port 3000 by default, Users can edit /etc/grafana/grafana.ini to
 modify the configuration

# Acccess grafana#using the default credentials admin / admin#you will be promoted to
 change the password on the first login.
http://{instance-ip}:3000/

# If you need public access, please configure an Application Load Balancer (ALB) on
 your own.
# When configuring the ALB, modify the Idle timeout to 1800 to avoid the following
 error during large data queries (when a single API call exceeds 60 seconds)#
# “a padding to disable MSIE and Chrome friendly error page”
```

Step 2: Authorize the EC2 where Grafana is located to access Athena

Prerequisite:

- You have deployed Grafana on EC2.
- EC2 has been configured with an IAM Instance Profile. You need to record the corresponding **role ARN** of the Instance Profile.

Follow the steps below:

1. Access IAM Management Console.
2. Search for the role including "`AthenaPublicAccessRole`" and choose it to access the details page. Record the role ARN, which will be used later.
3. Choose the **Trust relationships** tab.
4. Choose **Edit trust policy**.
5. Choose **Add** next to **Add a principal**.
6. Select **IAM Roles** from the **Principal type** drop-down list.
7. Enter the role ARN you have recorded in Step 2.
8. Choose **Add principal**.
9. Choose **update policy**.

Step 3: Install Amazon Athena plugins

Prerequisite:

- Grafana is installed.
- Grafana is accessible over the public network.

Follow the steps below:

1. Access the Grafana console.
2. Select **Administration** from the left navigation pane, and then choose **Plugins**.
3. Select **All** in the **State** section on the right side.
4. In the search box, enter `Athena` and choose the **Amazon Athena** to access the details page.
5. Choose **Install** on the page and wait for the plugin installation to complete.

Step 4: Create service accounts

Follow the steps below:

1. Access the Grafana console.
2. Select **Administration** from the left navigation pane, and then choose **Service accounts**.
3. Select **Add service account**.
4. Enter a display name. For example, "johndoe".
5. Select the role as `Admin`.
6. Choose **Create**.
7. Choose **Add service account token**.
8. Choose **Generate token**.
9. Choose **Copy to clipboard and close**.

10Save and record this token, which will be used when you need to create a pipeline.

OpenSSL 1.1 Installation

Centralized Logging with OpenSearch uses Fluent Bit as the log agent, which requires OpenSSL 1.1 or later. You can install the dependency according to your operating system (OS). It is recommended to make your own AMI with OpenSSL 1.1 installed.

> **Important**
>
> If your OS is not listed below, you can follow the official installation guide to install OpenSSL.

Amazon Linux 2

```
sudo yum install openssl11
```

Ubuntu

22.04

```
ln -s /usr/lib/x86_64-linux-gnu/libsasl2.so.2 /usr/lib/libsasl2.so.3
ln -s /snap/core18/current/usr/lib/x86_64-linux-gnu/libssl.so.1.1 /usr/lib/
libssl.so.1.1
ln -s /snap/core18/current/usr/lib/x86_64-linux-gnu/libcrypto.so.1.1 /usr/lib/
libcrypto.so.1.1
```

20.04

```
ln -s /usr/lib/x86_64-linux-gnu/libsasl2.so.2 /usr/lib/libsasl2.so.3
```

18.04

```
ln -s /usr/lib/x86_64-linux-gnu/libsasl2.so.2 /usr/lib/libsasl2.so.3
```

Debian

GNU/10

```
ln -s /usr/lib/x86_64-linux-gnu/libsasl2.so.2 /usr/lib/libsasl2.so.3
```

GNU/11

```
ln -s /usr/lib/x86_64-linux-gnu/libsasl2.so.2 /usr/lib/libsasl2.so.3
```

Red Hat Enterprise Linux

8.X

OpenSSL 1.1 is installed by default.

7.X

```
sudo su -

yum install -y https://s3.amazonaws.com/ec2-downloads-windows/SSMAgent/latest/
linux_amd64/amazon-ssm-agent.rpm

systemctl enable amazon-ssm-agent
systemctl start amazon-ssm-agent

yum install -y wget perl unzip gcc zlib-devel
mkdir /tmp/openssl
cd /tmp/openssl
wget https://www.openssl.org/source/openssl-1.1.1s.tar.gz
tar xzvf openssl-1.1.1s.tar.gz
cd openssl-1.1.1s
./config --prefix=/usr/local/openssl11 --openssldir=/usr/local/openssl11 shared zlib
make
make install

echo /usr/local/openssl11/lib/ >> /etc/ld.so.conf
ldconfig
```

SUSE Linux Enterprise Server

15

OpenSSL 1.1 is installed by default.

Upload SSL Certificate to IAM

Upload the SSL certificate by running the AWS CLI command upload-server-certificate similar to the following:

```
aws iam upload-server-certificate --path /cloudfront/ \
--server-certificate-name YourCertificate \
--certificate-body file://Certificate.pem \
--certificate-chain file://CertificateChain.pem \
```

```
--private-key file://PrivateKey.pem
```

Replace the file names and Your Certificate with the names for your uploaded files and certificate. You must specify the file:// prefix in the certificate-body, certificate-chain and private-key parameters in the API request. Otherwise, the request fails with a MalformedCertificate: Unknown error message.

> **Note**
>
> You must specify a path using the --path option. The path must begin with /cloudfront and must include a trailing slash (for example, /cloudfront/test/).

After the certificate is uploaded, the AWS command upload-server-certificate returns metadata for the uploaded certificate, including the certificate's Amazon Resource Name (ARN), friendly name, identifier (ID), and expiration date.

To view the uploaded certificate, run the AWS CLI command list-server-certificates:

```
aws iam list-server-certificates
```

For more information, see uploading a server certificate to IAM.

Developer guide

Visit our GitHub repository to download the source code for this solution. The solution template is generated using the AWS Cloud Development Kit (AWS CDK) (CDK). Refer to the README.md file for additional information.

Revisions

Date	Changes
March 2023	Initial release.
April 2023	Released version 1.0.1 Fixed deployment failure due to S3 ACL changes.
June 2023	Released version 1.0.3 Fixed the EKS Fluent Bit deployment configura tion generation issue.
August 2023	Released version 2.0.0 • Added feature of ingesting log from S3 bucket continuously or on-demand • Added log pipeline monitoring dashboard into the solution console • Supported one-click enablement of pipeline alarms • Added an option to automatically attach required IAM policies when creating an Instance Group • Displayed an error message on the console when the installation of log agent fails • Updated Application log pipeline creation process by allowing customer to specify a log source • Added validations to OpenSearch domain when importing a domain or selecting a domain to create log pipeline • Supported installing log agent on AL2023 instances

Date	Changes
	• Supported ingesting AWS WAF (associated with CloudFront) sampled logs to OpenSearch in other Regions except `us-east-1` • Allowed the same index name in different OpenSearch domains
September 2023	Released version 2.0.1 Fixed the following issues: • Automatically adjust log processor Lambda request's body size based on AOS instance type • When you create an application log pipeline and select Nginx as log format, the default sample dashboard option is set to "Yes" • Monitoring page cannot show metrics when there is only one dot • The time of the data point of the monitoring metrics does not match the time of the abscissa

Date	Changes
November 2023	Released version 2.1.0 • Added Light Engine to provide an Athena-based serverless and cost-effective log analytics engine to analyze infrequent access logs • Added OpenSearch Ingestion to provide more log processing capabilities, with which OSI can provision compute resource OpenSearch Compute Units (OCU) and pay per ingestion capacity • Supported parsing logs in nested JSON format • Supported CloudTrail logs ingestion from the specified bucket manually • Fixed the issue that the solution cannot list instances when creating instance groups • Fixed the issue that EC2 instances launched by the Auto Scaling group failed to pass the health check
December 2023	Released version 2.1.1 Fixed the following issues: • Instances should not be added to the same Instance Group • The solution cannot be deployed in United Arab Emirates (UAE) Region • Log ingestion error occurs in light engine when time key is not specified in the log config

Date	Changes
March 2024	Released version 2.1.2 Fixed the following issues: • The upgrade from versions earlier than 2.1.0 leads to the loss of Amazon S3 notifications, preventing the proper collection of logs from the Amazon S3 buffer • Including the "@timestamp" field in log configurations leads to failures in creating index_templates and an inability to write data to Amazon OpenSearch Service • Due to the absence of the 'batch_size' variable, process failures occur in the log processor Lambda function • The Log Analytics Pipeline could not deploy cross-account AWS Lambda pipelines • An issue with the ELB Service Log Parser resulted in the omission of numerous log lines • An inaccurate warning message is displayed during pipeline creation with an existing index in Amazon OpenSearch Service • Incorrect error message occurs when deleting an instance group in Application Logs
May 2024	Documentation update: Updated and corrected AWS service names.

Contributors

- Chang, Owen
- Chen, Haiyun
- Dai, Aiden
- Grover, Lalit
- Hu, Yikai
- Liu, Eva
- Luo, Robin
- Ma, James
- Qian, Yang
- Shi, Joe
- Wang, Haidong
- Wei, Charles
- Xu, Ming

Notices