Implementation Guide

Migration Assistant for Amazon OpenSearch Service

Migration Assistant for Amazon OpenSearch Service: Implementation Guide

Table of Contents

Build an environment to upgrade, migrate, and compare OpenSearch clusters

Publication date: *November 2023 (last update: December 2024)*

OpenSearch is widely adopted for log analytics and search functionalities. However, self-managing OpenSearch can be operationally demanding. Amazon OpenSearch Service and Amazon OpenSearch Service Serverless offer more manageable alternatives, but transitioning to these services or updating to the latest OpenSearch version has historically been complex. Also, it can be difficult for a customer to predict the outcome of a migration. The Migration Assistant for Amazon OpenSearch Service solution addresses these challenges - it simplifies the migration process, ensures integrity, and validates performance post-migration.

The Migration Assistant for Amazon OpenSearch Service solution is a toolkit designed to ease the transition to OpenSearch, facilitate upgrades to the latest OpenSearch versions, and refine cluster configurations based on observed traffic patterns. Whether you're looking to set up a proof-of-concept in AWS, transition production workloads with confidence, or enhance your current OpenSearch clusters, this guide provides references to step-by-step instructions, best practices, and insights to leverage the full potential of the OpenSearch migrations package.

Benefits of using this solution:

- Migrate cluster metadata, including index settings, type mappings, index templates, and aliases.
- Migrate existing data from legacy clusters to OpenSearch clusters, including Amazon OpenSearch Service (AOS) Domains and Amazon OpenSearch Service Serverless (AOSS) collections.
- Intercept and redirect live traffic from self-managed Elasticsearch or OpenSearch clusters to Amazon OpenSearch Service domains and Amazon OpenSearch Service Serverless collections with minimal latency.
- Replicate production traffic on target clusters to validate and ensure accuracy.
- Simulate real-world traffic by capturing and replaying request patterns to fine-tune system performance.
- Deploy across the most common AWS Regions for global reach and scalability.
- Provides a recommended path for migration while continuing to maintain service availability.

Note

The Migration Assistant for Amazon OpenSearch Service solution supports migrating from Elasticsearch versions 6.x and 7.x, and OpenSearch 1.x and 2.x. The following migrations paths are tested regularly:

- Elasticsearch 6.8 to OpenSearch 2.14
- Elasticsearch 7.10.2 to OpenSearch 2.14
- Elasticsearch 7.17 to OpenSearch 2.14
- OpenSearch 1.3 to OpenSearch 2.14

We expect other minor versions within these major versions to work. If you prefer another migration path or find a bug with an existing path, open a feature request or bug accordingly.

This implementation guide provides an overview of the Migration Assistant for Amazon OpenSearch Service solution, its reference architecture and components, considerations for planning the deployment, and configuration steps for deploying the solution to the Amazon Web Services (AWS) Cloud. It also references the solution's open-source documentation on GitHub, which includes a User guide, developer documentation, and tips to enhance and contribute to the solution.

The intended audience for using this solution's features and capabilities in their environment includes solution architects, business decision makers, DevOps engineers, data scientists, and cloud professionals.

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

If you want to . . .	Read . . .
Know the cost for running this solution. The estimated cost for running this solution in the US East (N. Virginia) Region is approxima tely USD $3,096 for a 15-day migration with	Cost

If you want to . . .	Read . . .
100 TB of existing data and 15 MBps of live traffic.	
Understand the security considerations for this solution.	Security
Know how to plan for quotas for this solution.	Quotas
Know which AWS Regions support this solution.	Supported AWS Regions
View or download the AWS CloudForm ation template included in this solution to automatically deploy the infrastructure resources (the "stack") for this solution.	AWS CloudFormation template
Access the source code and optionally use the AWS Cloud Development Kit (AWS CDK) to deploy the solution.	GitHub repository

Features and benefits

The solution provides the following features:

Backfill with reindex-from-snapshot

This solution guides users through the process of transferring data from a snapshot stored in an Amazon Simple Storage Service (Amazon S3) bucket to a designated (target) cluster.

Live traffic capture and replay

The solution offers guidance and tools to intercept traffic intended for an original cluster and archive it for future replay on a destination cluster. Typically, the replay occurs at the same rate and concurrency as the original traffic to precisely mimic the workload experienced by the source cluster. Users can choose to replay the recorded traffic subsequently or adjust the replay speed. This flexibility enables users to fine-tune the target cluster, enhancing its performance to suit their requirements.

Traffic validation

The solution records requests and responses between the source and destination clusters for comparison. It then forwards the latency metrics and response codes to an analytics platform, enabling users to analyze the data essential for transitioning their traffic from a legacy system to a new Amazon OpenSearch Service destination. The solution stores each request and response for both the source and target for deeper inspection if needed.

Integration with AWS Service Catalog AppRegistry and Application Manager, a capability of AWS Systems Manager

This solution includes a Service Catalog AppRegistry resource to register the solution's CloudFormation template and its underlying resources as an application in both Service Catalog AppRegistry and Application Manager. With this integration, you can centrally manage the solution's resources and enable application search, reporting, and management actions.

Use cases

Migrating existing data

Migration Assistant for Amazon OpenSearch Service offers various options for migrating existing data, including detailed guidance on running a reindex-from-snapshot applicable across all supported migration routes, such as from Elasticsearch 6.8, 7.10.2, or 7.17 to OpenSearch 2.14.

Near real-time migration of HTTP traffic between clusters

The solution offers you the option to capture data destined for a source cluster and store this data for reuse. A user can replay this data to a target cluster in near real-time to migrate as soon as possible, or replay at a later time.

Replay traffic to multiple targets

The solution allows you to capture traffic for replay through multiple instances or in sequential runs, facilitating the validation of diverse cluster workloads and configurations.

Precise simulation of your cluster workloads

The solution allows users to capture and replay traffic either simultaneously with multiple instances, or in separate sequential runs. This feature aids in validating different cluster workloads and configurations. By default, the Traffic Replayer preserves the original concurrency and request rate to accurately simulate production loads, ensuring a fair like-for-like comparison.

Validate target cluster results

The solution facilitates user comparisons of source and target traffic in terms of accuracy and performance. It captures metrics and logs for analysis, providing users with the necessary confidence to migrate their production traffic to a new target.

Concepts and definitions

This section describes key concepts and defines terminology specific to this solution:

source cluster

The originating cluster on a specific version of Elasticsearch or OpenSearch that the user is attempting to either upgrade or decommission.

target cluster

The destination cluster that the user is trying upgrade, migrate to, or optimize.

capture proxy

A pass-through HTTP proxy designed to capture and log all of the request and response traffic to a durable source for later reuse.

Traffic Replayer

A tool designed to simulate original traffic workloads by retrieving recorded request traffic and sending it to a target cluster. The Traffic Replayer correlates the request and response traffic of the originating request with the request and response traffic to the target, and stores the traffic persistently.

existing data

Documents that were on the source cluster at the point where a snapshot is taken.

live/continuous data

Data intercepted by the Capture Proxy and subsequently processed through a Traffic Replayer. Initially, this information is transmitted from clients to the source cluster, where it is intercepted by the Capture Proxy. Subsequently, the data is relayed back to the designated target cluster.

Note

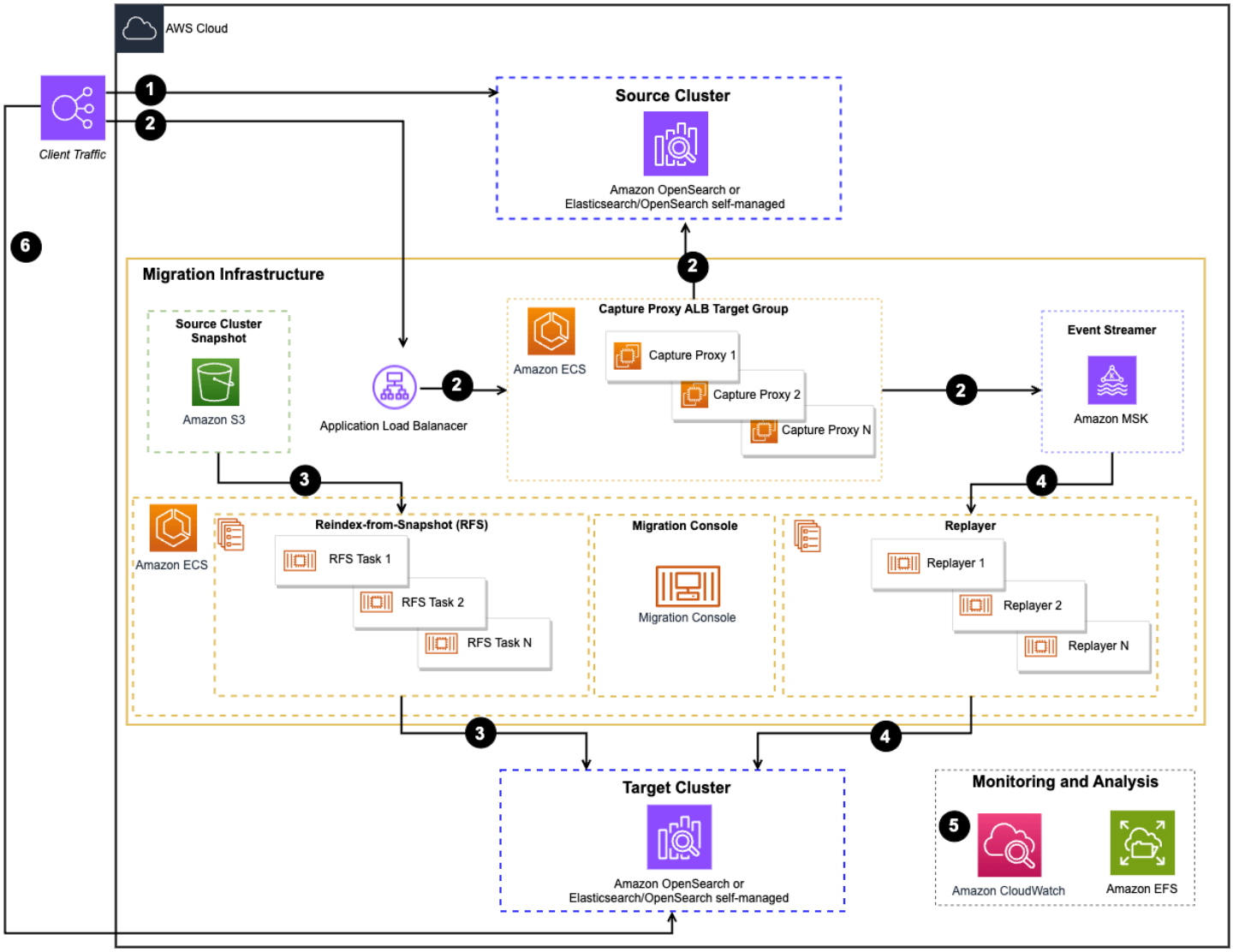
For a general reference of AWS terms, see the AWS Glossary.

Architecture overview

This section provides a reference implementation architecture diagram for the components deployed with this solution.

Architecture diagram

Deploying this solution with the default parameters deploys the following components in your AWS account.



Migration Assistant for Amazon OpenSearch Service architecture on AWS

> **Note**
>
> AWS CloudFormation resources are created from AWS Cloud Development Kit (AWS CDK) constructs.

The high-level process flow for the solution components deployed with the AWS CloudFormation template is as follows:

1. Client traffic is directed to the existing cluster.
2. An Application Load Balancer is positioned in front of the traffic capture proxy to route traffic as needed. The Application Load Balancer forwards traffic to the capture proxy, which then relays it to the source while simultaneously replicating the traffic to Amazon Managed Streaming for Apache Kafka (Amazon MSK).
3. With continuous traffic capture in place, a Reindex-from-Snapshot (RFS) is initiated by the user through the Migration Management Console.
4. After a backfill has been completed, the captured traffic is replayed by the user with a Traffic Replayer.
5. The performance and behavior of traffic routed to the source and target clusters are analyzed by reviewing relevant logs and metrics.
6. After confirming the target cluster's functionality meets expectations, the user redirects clients to the new target. Additionally, the user can retire and discard the old cluster's infrastructure.

AWS Well-Architected design considerations

This solution uses the 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 benefit this solution.

Operational excellence

This section describes how we architected this solution using the principles and best practices of the operational excellence pillar.

- The solution enables users to transition from legacy, self-managed OpenSearch and OpenSearch clusters to AWS OpenSearch, a managed service that significantly reduces the operational burden associated with maintaining outdated systems.

Security

This section describes how we architected this solution using the principles and best practices of the security pillar.

- The solution facilitates user migration to Amazon OpenSearch Service, which is regularly updated to uphold a strong security posture.

Reliability

This section describes how we architected this solution using the principles and best practices of the reliability pillar.

- The solution introduces a lightweight proxy layer between the client and the source cluster to capture data reliably using Kafka. This ensures that requests can be replayed on the target cluster, with the flexibility to modify or resend any failed requests.

Performance efficiency

This section describes how we architected this solution using the principles and best practices of the performance efficiency pillar.

- The tools within the solution have been optimized for efficiently capturing and replaying traffic, mitigating the need for horizontal scaling.

Cost optimization

This section describes how we architected this solution using the principles and best practices of the cost optimization pillar.

- The solution provisions CloudFormation stacks designed to operate specifically during migration processes, which can be safely decommissioned post-migration.

- The solution operates on AWS Fargate within Elastic Container Service, ensuring cost-effectiveness as charges are incurred only for the duration that the containers are active.
- For those interested in a preliminary local trial, a local version of the solution can be built and deployed in a user's own environment before cloud implementation. Detailed instructions and resources for this are available at the OpenSearch Migrations GitHub repository: https://github.com/opensearch-project/opensearch-migrations.

Sustainability

This section describes how we architected this solution using the principles and best practices of the sustainability pillar.

- The solution allows customers to migrate their workloads to more environmentally friendly hardware, for example Graviton instance types.

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. These components are meant to accomplish one of the following scenarios:

- **Metadata migration** - Migrating cluster metadata, such as index settings, aliases, and templates.
- **Backfill migration** - Migrating existing or historical data from a source to a target cluster.
- **Live traffic migration** - Replicating live ongoing traffic from source to target cluster.
- **Comparative tooling** - Comparing the performance and behaviors of an existing cluster with a prospective new one.

In this guide, we focus on the first three scenarios, guiding you through a backfill from a source cluster while concurrently handling live production traffic, which will be captured and replayed to a target cluster.

> **Important**
>
> Migration strategies aren't universally applicable. This guide provides instructions based on engineering best practices.

AWS services in this solution

AWS service	Description
AWS CloudFormation	**Core.** Infrastructure as Code (IaC) templates used to deploy and configure Migration Assistant.
Amazon OpenSearch Service (AOS)	**Core.** A Search, Logging, and Analytics Engine that users can upgrade to, migrate to, and use to compare the results of a source and target cluster.

AWS service	Description
Amazon Managed Streaming Service for Apache Kafka (MSK)	**Core.** Stream-processor that is fully managed. It is used as a durable way to store and reuse HTTP traffic.
Amazon Elastic Container Service (ECS)	**Core.** Runs highly secure, reliable, and scalable containers. The Migration Management Console and Traffic Replayer run in Amazon ECS.
Amazon Elastic File System	**Core.** Scalable persistent storage utilized for retaining the request and response data from both the source and target clusters.
Amazon S3	**Core.** Storage allocated for Historical Backfill tasks, which involves exporting a snapshot from the source to be restored by the target cluster. S3 is also used to store IaC content.
AWS Systems Manager	**Supporting.** Provides you visibility and control of your infrastructure on AWS. Systems Manager provides a unified user interface so you can view operational data from multiple AWS services and enables you to automate operational tasks across your AWS resources.
AWS Secrets Manager	**Supporting.** A secure way for storing sensitive data, such as cluster credentials, that is required for Migration Assistant.
Amazon EC2	**Supporting.** Provides networking and security infrastructure for Migration Assistant including securing groups, and Virtual Private Networks.

AWS service	Description
AWS Lambda	**Supporting.** Lambda facilitates the execution of serverless functions and is employed by Migration Assistant to operate its suite of tools.
Amazon CloudWatch	**Optional.** Observe and monitor resources and applications on AWS or in the local Docker solution.

Additionally, the following tools are used in this solution:

Service	Description
Amazon OpenSearch Serverless	**Supporting.** A serverless version of OpenSearch, which functions as a search, logging, and analytics engine, offers users the flexibility to migrate to and compare as the target cluster. Configuration of this target is at the user's discretion.

Self-service Elasticsearch/OpenSearch Source Cluster

The source cluster for this solution is based on Elasticsearch or OpenSearch, operating on EC2 instances or alternative computing infrastructure. Configure a capture proxy to interface with the source cluster, positioning the proxy in front of, or on each of the cluster coordinating nodes.

Migration Management Console

The Migration Management Console is a containerized portal that operates on Fargate within Amazon Elastic Container Service (Amazon ECS). Its primary role is to facilitate running the Migration Assistant for Amazon OpenSearch Service solution, along with providing a suite of tools designed to aid in the migration process. The console provides a migration-specific CLI and offers a variety of tools to streamline the migration process. Everything necessary for completing a migration, other than cleaning up the migration resources, can be done through this console.

Metadata Migration Tool

The Metadata Migration Tool is integrated into the Migration Management Console CLI. You can also use the Metadata Migration Tool to migrate cluster metadata, including index mappings, index configuration settings, templates, component templates, and aliases.

Capture Proxy

Capture Proxy is designed for HTTP RESTful traffic. It functions by relaying traffic to a source cluster and concurrently dividing the traffic, replicating it into a durable Kafka stream for subsequent playback.

Traffic Replayer

Traffic Replayer is a network traffic utility that replicates real-world workloads by retrieving recorded request traffic and dispatching it to a designated target cluster. It associates the original requests and their responses with those directed to the target cluster. This helps to compare correlated data.

Reindex-from-Snapshot Container

The Reindex-from-Snapshot (RFS) Container reindexes data from an existing snapshot on Amazon ECS tasks that coordinate the migration of documents from an existing snapshot. This reindexes documents in parallel to a target cluster.

Target Cluster

The Target Cluster is the destination OpenSearch cluster for migration or comparison in an A/B test. This component must exist prior to deploying this solution.

How this solution works

The Migration Assistant for Amazon OpenSearch Service solution simplifies zero-downtime migrations by capturing, replicating, and replaying traffic between source and target clusters. Client traffic initially flows to the existing source cluster, with an Application Load Balancer directing it to a traffic capture proxy. This proxy forwards traffic to the source while replicating it to Amazon Managed Streaming for Apache Kafka (Amazon MSK) for later replay.

After the traffic capture is active, users initiate a Reindex-from-Snapshot (RFS) process through the Migration Management Console to backfill existing data. After the backfill completes, users employ the Traffic Replayer to synchronize live traffic on the target cluster. Logs and metrics are analyzed to compare the performance and behavior of the source and target clusters, helping ensure that the new setup is functioning as expected. After confirmed, client traffic is redirected to the target cluster, allowing for the safe retirement of the old infrastructure.

This solution supports several scenarios, including metadata migration, backfilling historical data, and replicating live traffic. With tools such as the Traffic Capture Proxy, Traffic Replayer, and Metadata Migration Tool, users can efficiently manage the migration process, helping to ensure that the target environment mirrors the source without disruptions to live services.

Application Load Balancer

An Application Load Balancer manages and distributes incoming network traffic across multiple targets, such as servers or containers. The Migration Assistant Application Load Balancer is configured to manage traffic for different components of our migration system. The solution uses this Application Load Balancer to distribute traffic to the proxy during the live capture phase of the migration, and to eventually direct traffic to the target cluster.

For more details on how the capture proxy works to shift client traffic during phases of the migration, refer to the Client Traffic Switchover repository documentation.

Metadata Migration Tool

The Metadata Migration Tool migrates templates and indices in a specified snapshot of a source cluster to a target cluster. It parses the contents of the snapshot to extract the settings and configuration of the templates, aliases, and indices in the snapshot. Then it migrates the settings and configuration to the target cluster. The user can apply allowlists to filter which templates and

indices are migrated. If a template or index of the same name already exists on the target cluster, this tool doesn't overwrite the existing one on the target. Instead, the tool leaves the pre-existing index and template settings as-is without modifications.

The tool also applies basic transformations to the template and index settings to handle upgrades between version-specific behavior. See the Metadata Migration Tool README for more information.

Traffic Capture and Replay

The Capture Proxy relays network traffic from HTTP requests into a durable, scalable stream before forwarding them to the source cluster. The Traffic Replayer replicates the captured traffic through a Kafka topic as it also sends the requests to the downstream source cluster. The Traffic Replayer then pulls the Kafka records, reconstructs them into HTTP requests, and sends them to the target cluster. This replication can test the performance of the new target cluster, keep the clusters synchronized, and illustrate differences between a source and a target. Users can explore the progress and performance of the replay through CloudWatch metrics and through the Migration Management Console by querying results, which are output as rotating JSON files.

For more details about the implementation of the Capture Proxy and Traffic Replayer, refer to the Traffic Capture and Replay design document and the README files for the Capture Proxy and the Traffic Replayer.

Backfill migration

Backfill is accomplished with RFS by moving existing documents in your source cluster to the target cluster. The RFS takes a snapshot of a source cluster. Then a separate process (or processes) parses the snapshot to extract the documents and reindex them on the target cluster. This process doesn't involve the source cluster again after the snapshot is finished. The format of Elasticsearch and OpenSearch indices is such that each shard of each index can be parsed, extracted, and reindexed independently, meaning the work can be fanned out at the shard level.

This approach improves the migration experience by:

- Removing load from the source cluster during backfill migration after the Source Cluster Snapshot is taken
- Enabling "hopping" across multiple major versions without having to pass through the intermediate versions
- Creating a migration path from post-fork versions of Elasticsearch to OpenSearch

- Increasing the speed of backfill migration by parallelizing work at the shard level
- Simplifying the process of pausing and resuming a migration

For more context, refer to RFC Issue 12667 and the RFS High Level Design.

Plan your deployment

This section describes the Region, cost, security, and quota considerations for planning your deployment.

Supported AWS Regions

This solution uses Amazon OpenSearch Service, which is not currently available in all AWS Regions. For the most current availability of AWS services by Region, see the AWS Regional Services List.

Migration Assistant for Amazon OpenSearch Service is available in the following AWS Regions:

Region name	
US East (N. Virginia)	Asia Pacific (Tokyo)
US East (Ohio)	Europe (Frankfurt)
US West (Oregon)	Europe (Ireland)
US West (N. California)	Europe (London)
Asia Pacific (Singapore)	AWS GovCloud (US-East)
Asia Pacific (Sydney)	AWS GovCloud (US-West)

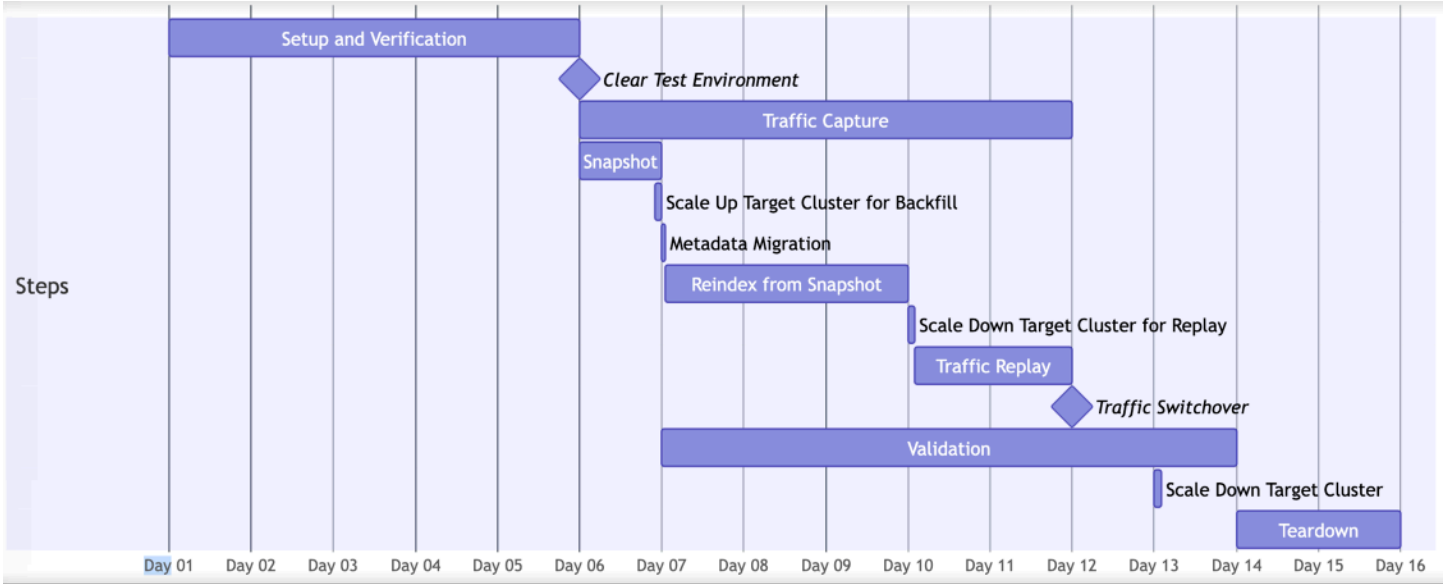
Cost

You are responsible for the cost of the AWS services used while running this solution. As of this revision, the cost for running this solution with the default settings in the US East (N. Virginia) Region is approximately **$3,096 for a 15-day migration with 100 TB of existing data and 15 MBps of live traffic**. These costs are for the resources shown in the sample cost table.

We recommend creating a budget through AWS Cost Explorer to help manage costs. Prices are subject to change. For full details, refer to the pricing webpage for each AWS service used in this solution.

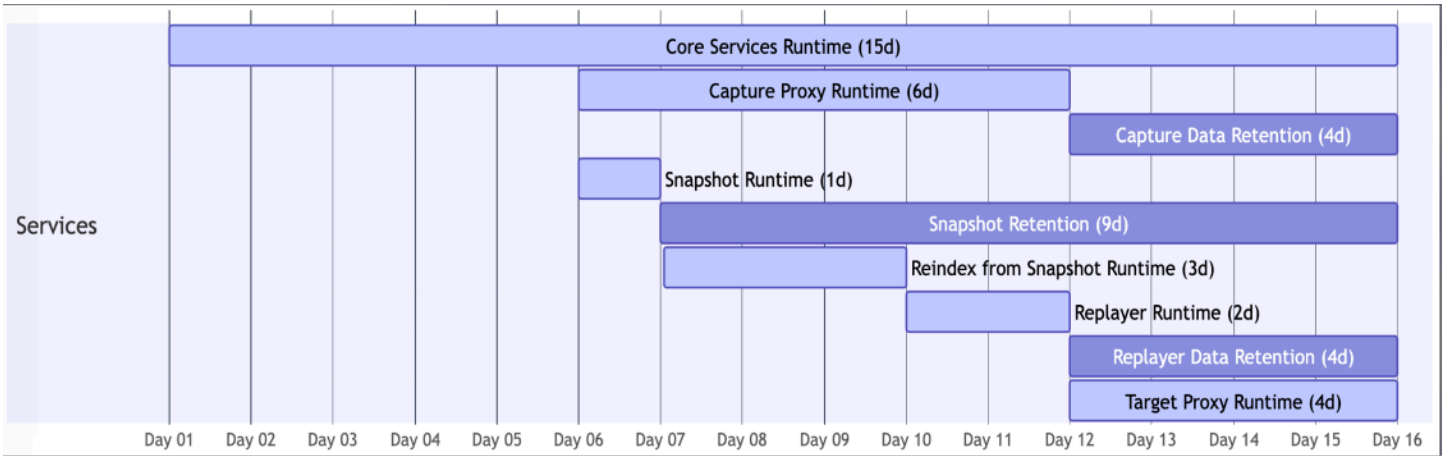
A migration typically has different duration and data volume for the individual steps described in the OpenSearch Migrations Wiki. Customers typically wait to remove infrastructure or delete data until after a migration is complete. It's crucial to understand the volume and duration of each of the steps to estimate the cost of the solution.

In the following example, we outline the cost of a 15-day migration with the following schedule:



Example 15-day migration schedule

To understand the costs, we need to map these steps into cost components, including data retention periods as applicable. This yields the following schedule:



Example 15-day migration schedule mapped to solution components

- Core Services – 15 days
- Capture Proxy – 6 days

- Capture Proxy Data Retention Period – 4 days
- Snapshot – 1 day
- Snapshot Data Retention Period – 9 days
- Reindex from Snapshot – 3 days
- Traffic Replayer – 2 days
- Traffic Replayer Data Retention Period – 4 days
- Target Proxy – 4 days

Using this schedule, the following is an example of a customer performing a zero-downtime migration of a cluster with 100 TB of primary shard data with 15 MBps aggregated request-response throughput. The ongoing request throughput over the 6-day capture timeline gives a capture/replay volume of about 7.8 TB.

Assumptions:

- No data is retained after the teardown step, representing no resources and cost after day 15.
- Deployment in US East (N. Virginia).
- The Source Cluster, Target Cluster, and VPC are imported and cost excluded from this calculation, including data transfer cost based on VPC setup. We recommend that the VPC includes an Amazon S3 Gateway Endpoint and cluster data communication over VPC Interface Endpoints to reduce data transfer cost.

This gives us the following cost table:

AWS service	Dimensions	Cost [USD]
Core services		
Amazon ECS	Migration Management Console: Task vCPU: *1 task × 24 hours per day × 15 days × (0.50 vCPU × 0.04048 USD per hour) = $7.29*	$8.89

AWS service	Dimensions	Cost [USD]
	Task Memory: *1 task × 24 hours per day × 15 days × (1 GB × 0.004445 USD per hour) = $1.60*	
AWS Lambda	*< 10 requests × 0.20 per 1M requests = ~$0.00* *< 10 seconds × $0.000016 6667 per GB-second = ~$0.00*	~$0.00
AWS Secrets Manager	*($0.40 per secret-month × 10 secrets × 15 days) / 30 days per month = $2.00*	$2.00
Amazon Route 53	*($0.50 per Hosted Zone × 15 days) / 30 days per month = $0.25*	$0.25
Amazon Elastic Container Registry (Amazon ECR)	*(3 GB × $0.10 GB/month × 15 days) / 30 days per month = $0.15*	$0.15
Amazon EC2	Bootstrap Box (t3.large): *1 × $0.0832/hr × 24 hours per day × 15 days = $29.95*	$29.95
Amazon EBS	Bootstrap Box: *(50 GB × $0.08/GB-month × 15 days) / 30 days per month = $2.00*	$2.00

AWS service	Dimensions	Cost [USD]
Amazon Virtual Private Cloud (Amazon VPC)	Bootstrap Box VPC – NAT Gateway: *1 × 24 hours per day × 15 days × $0.045/hour = $16.20* *30 GB × $0.045/GB = $1.35*	$17.55
Elastic Load Balancing	Application Load Balancer hours: *15 days × 24 hours × $0.0225 per Application Load Balancer-hour = $8.10*	$8.10
Core services total:		**$68.89**
Miscellaneous services		

AWS service	Dimensions	Cost [USD]
Amazon CloudWatch	**Note** This is a high estimate for aggregated charges over all components for a large migration. Log data ingested (200 GB): *$0.50 per GB × 200 GB = $100.00* Archived log charges (assume log data compresses to 30 GB): *($0.03 per GB-month × 30 GB × 15 days) / 30 days per month = $0.45* Metrics (200 metrics): *(200 metrics × $0.30 per metric-month × 15 days) / 30 days per month = $60.00* **Note** Metrics are metered only while being sent. Actual usage and cost might be lower.	$160.45

AWS service	Dimensions	Cost [USD]
AWS X-Ray	*1M traces × $5.00 per million traces = $5.00*	$5.00
Miscellaneous services total:		**$165.45**
Optional services		
Capture proxy (7.776 TB), 6 days running; 4 days data retention)		
Amazon Managed Streaming for Apache Kafka (Amazon MSK)	**Note** Assume deployment in 2 Availability Zones. Broker (6 days running + 4 days data retention): *2 nodes (M5.large) × $0.21 node per hour × 10 days × 24 hours per day = $151.20* Storage (9,331 GB Provision ed (20% buffer) for 6 days running + 4 days data retention): *(2 nodes × 9,331 GB × $0.10 per GB-month × 10 days) / 30 days per month = $620.07* Data transfer: *7,776 GB × $0.01 per GB inbound = $77.76*	$849.03

AWS service	Dimensions	Cost [USD]
Amazon ECS	Task vCPU: *4 tasks × 0.5 vCPU × 0.04048 USD per vCPU-hour × 6 days × 24 hours per day = $11.66* Task memory: *4 tasks × 2 GB × 0.004445 USD per GB-hour × 6 days × 24 hours per day = $5.12*	$16.78
Elastic Load Balancing	Load Balancer Capacity Unity (LCU) (assuming GB is the cost dimension): *7,776 GB × $0.008 per LCU = $62.21*	$62.21
Capture proxy total:		**$928.02**
Target proxy (15 MBps for 4 days (5.184 TB))		
Amazon ECS	Task vCPU: *4 tasks × 0.5 vCPU × 0.04048 USD per vCPU-hour) × 4 days × 24 hours per day = $7.77* Task memory: *4 tasks × 2 GB × 0.004445 USD per GB-hour × 4 days × 24 hours per day = $3.41*	$11.18

AWS service	Dimensions	Cost [USD]
Elastic Load Balancing	LCU (assuming GB is the cost dimension): *5,184 GB × $0.008 per LCU = $41.47*	$41.47
Target proxy total:		**$52.65**
Replay (7.776 TB over 2 days, retention 4 days)		
Amazon ECS	Task vCPU: *1 task × 1 vCPU × 0.04048 USD per vCPU-hour × 2 days × 24 hours per day = $1.94* Task memory: *1 task × 4 GB × 0.004445 USD per GB-hour × 2 days × 24 hours per day = $0.86*	$2.80
Amazon MSK	Intra-Region: *7,776 GB × 0.01 USD per GB outbound = $77.76*	$77.76

AWS service	Dimensions	Cost [USD]
Amazon Elastic File System (Amazon EFS)	**Note** Calculating cost based on constant size of 15.552 TB (7.776 TB × 2). Actual cost might differ due to some data spending less than 5 days in EFS Infrequent Access or data inflation. EFS Standard (infrequent after 1 day): *15,552 GB × $0.30 per GB-month × 1 day / 30 days per month = $155.52* EFS Infrequent Access (5 days maximum): *15,552 GB × $0.025 per GB-month × 5 days / 30 days per month = $64.80* EFS Infrequent Access – Tiering: *15,552 GB × $0.01 per GB = $155.52*	$375.84
Replay total:		**$456.40**
Historical Backfill (100 TB Snapshot over 1 month)		

AWS service	Dimensions	Cost [USD]
Amazon ECS	RFS: Each task is capable of delivering approximately 5 MBps of snapshot data. To deliver 100 TB over 3 days, 77 tasks are needed. Task vCPU: *77 tasks × 3 days × 24 hours per day × 2 vCPUs × $0.04048 per hour = $448.84* Task memory: *77 tasks × 3 days × 24 hours per day × 4 GB × $0.004445 per hour = $98.57* Task storage: *77 tasks × 3 days × 24 hours per day × (200GB - 20GB Free Tier) × $0.000111 per hour = $110.77*	$657.34

AWS service	Dimensions	Cost [USD]
Amazon S3	**Note** This calculates the cost for all the data occurring in the S3 bucket for the entire duration. The actual cost might be less for storage due to partial rate while the snapshot is being taken. Snapshot storage: *(100 TB × 1,000 GB per TB × $0.023 per GB-month × 10 days) / 30 days per month = $766.66*	$766.66
Historical Backfill total:		**$1,424.00**
Total for all components:		**$3,095.41**

The Capture and Replay yields an effective cost per TB of $199.87, and the Historic Backfill is $15.41 per TB. For a given cluster, the amount of data in Historic Backfill is largely fixed, while the data for Capture and Replay is based on the time needed to capture. For this reason, it can be cost advantageous to scale up the target OpenSearch cluster beyond the final intended capacity for the Historic Backfill period to reduce the duration and quantity of data for Capture and Replay.

Taking the table above and dividing out the components cost, we can determine the following calculation method for cost. This is a broad estimation making assumptions on traffic pattern including size per request, and number of new connections. We can also subdivide this by migration type applicable.

Component	Applicable migration type	Cost [USD]
Core services	• Metadata only • Historical only • Live only • Full migration	$4.59/day
Miscellaneous services	• Metadata only • Historical only • Live only • Full migration	$165.45 per large migration
Capture runtime	• Live only • Full migration	$17.22/day + $8.00/TB-day + $18.00/TB
Capture data retention period	• Live only • Full migration	$15.12/day + $8.00/TB-day
Snapshot	• Historical only • Full migration	$0.7667/TB-day
Snapshot data retention period	• Historical only • Full migration	$0.7667/TB-day
Reindex from snapshot	• Historical only • Full migration	$6.57/TB
Traffic Replayer	• Live only • Full migration	$1.40/day + $50.00/TB + $1.6667/TB-day **Note** Due to Amazon EFS-Intelligent Tiering,

Component	Applicable migration type	Cost [USD]
		$1.6667/TB-day is not paid for the first day.
Traffic Replayer data retention period	• Historical only • Full migration	$1.6667/TB-day
Target proxy	• Historical only • Full migration	$2.10/day + $8.00/TB

We can also identify the cost for full migrations of different sizes that follow the 15-day migration timeline. By applying the durations shown previously, as well as a conversion from TB/day to MB/s, we get the following formula for a representative cost estimate:

15-day migration = $390.35 + $14.24 × HistoricalTB + $71.86 × LiveMBps

This results in the following cost estimates table:

Note

All costs are rounded to the nearest dollar.

15-day full migration cost ($ USD/migration)			
	Live traffic throughput		
Historical data volume	5 MBps	20 MBps	50 MBps
1 TB	$764	$1,842	$4,125
10 TB	$892	$1,970	$3,252
100 TB	$2,174	$3,252	$5,407

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, visit 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 aims to create IAM roles with least privilege where resource access is required. This includes allowing some required Migration ECS services to produce/consume from MSK, make requests to the target cluster, and access provided secrets stored within AWS Secrets Manager needed for target cluster authentication and authorization.

Security groups

The solution creates security groups designed to control and isolate network traffic between Migration ECS containers, as well as between certain Migration ECS containers and associated services such as Amazon MSK, Amazon OpenSearch Service, and Amazon EFS. We recommend that you review the security groups and further restrict access as needed once the deployment is up and running.

AWS Secrets Manager

Migration Assistant for Amazon OpenSearch Service allows accessing stored secrets from AWS Secrets Manager in the Migration Management Console, Traffic Replayer, and Reindex-from-Snapshot containers. Accessing these secrets allows for proper authentication when migrating data from source to target, and for observing migration status.

Quotas

Service quotas, also referred to as limits, are the maximum number of service resources or operations for your AWS account.

Quotas for AWS services in this solution

Make sure you have sufficient quota for each of the services implemented in this solution. For more information, refer to AWS service quotas.

Use the following links to go to the page for that service. To view the service quotas for all AWS services in the documentation without switching pages, view the information in the Service endpoints and quotas page in the PDF instead.

AWS CloudFormation quotas

Your AWS account has CloudFormation quotas that you should be aware of when launching the stack for this solution. By understanding these quotas, you can avoid limitation errors that would prevent you from deploying this solution successfully. For more information, refer to AWS CloudFormation quotas in the *AWS CloudFormation Users Guide*.

Deploy the solution

This solution uses AWS CloudFormation templates and stacks to automate its deployment. The CloudFormation template specifies the AWS resources included in this solution and their properties. The CloudFormation stack provisions the resources that are described in the template.

Important

To update to a newer version of this solution, uninstall and redeploy it. This solution isn't backwards-compatible.

Deployment process overview

Follow the step-by-step instructions in this section to configure and deploy the solution into your account.

Before you launch the solution, review the cost, architecture, network security, and other considerations discussed earlier in this guide.

Time to deploy: Approximately 45-60 minutes (with MSK requiring the bulk of time)

Step 1: Launch the bootstrap stack

Step 2: Setup the bootstrap instance

Step 3: Customize the migration options

Step 4: Deploy the migration stacks

Important

This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. AWS owns the data gathered though this survey. Data collection is subject to the Privacy Notice.

To opt out of this feature, download the template, modify the AWS CloudFormation mapping section, and then use the AWS CloudFormation console to upload your updated

template and deploy the solution. For more information, see the Anonymized data collection section of this guide.

AWS CloudFormation template

You can download the CloudFormation template for this solution before deploying it.

View template

migration-assistant-for-amazon-opensearch-service-create-vpc.template - Use this template to launch the solution and all associated components. This configuration handles creating a VPC, which solution resources will be placed into, as well as a bootstrap EC2 instance, which will be accessed for customizing and deploying the needed migration resources for the solution.

View template

migration-assistant-for-amazon-opensearch-service-import-vpc.template - Use this template to launch the solution and all associated components. This configuration allows importing an existing VPC, which solution resources will be placed into, as well as creating a bootstrap EC2 instance, which will be accessed for customizing and deploying the needed migration resources for the solution.

Note

AWS CloudFormation resources are created from AWS Cloud Development Kit (AWS CDK) constructs.

Step 1: Launch the bootstrap stack

Follow the step-by-step instructions in this section to configure and deploy the bootstrap stack into your account.

Time to deploy: Approximately 5 minutes

1. Sign in to the AWS Management Console and select the button to launch the `migration-assistant-for-amazon-opensearch-service.template` CloudFormation template.

OR

Sign in to the AWS Management Console and select the button to launch the `migration-assistant-for-amazon-opensearch-service-import-vpc.template` CloudFormation template.

2. The template launches in the US East (N. Virginia) Region by default. To launch the 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 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, see IAM and AWS STS quotas, name requirements, and character limits in the *AWS Identity and Access Management User Guide*.
5. Under **Parameters**, review the parameters for this solution template and modify them as necessary. This solution uses the following default value.

   If you are launching the solution in a new VPC, this solution uses the following parameters:

Parameter	Default	Description
Stage	`dev`	Specify the stage identifier which will be used in naming resources, for example, `dev`, `gamma`, `wave1`.

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

Parameter	Default	Description
Stage	`dev`	Specify the stage identifier which will be used in naming resources, for example, `dev`, `gamma`, `wave1`.
VPC	`<required>`	Select a VPC. We recommend choosing the VPC of the target cluster.
Availability Zones	`<required>`	Select Availability Zones in the selected VPC. Please provide two zones at least, corresponding with the private subnets selected next.
Private Subnets	`<required>`	Select Private Subnets in the selected VPC. Please provide two subnets at least, corresponding with the availability zones selected previously.

6. Select **Next**.
7. On the **Configure stack options** page, choose **Next**.
8. On the **Review and create** page, review and confirm the settings. Select the box acknowledging that the template might create IAM resources.
9. Choose **Submit** 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.

Step 2: Setup the bootstrap instance

1. From the local environment where you will access the bootstrap instance, configure the required AWS credentials to allow access to the bootstrap instance. The identity used must have permissions that allow `ssm:StartSession` on the deployed bootstrap instance and SSM document resource.

   > **Note**
   >
   > We recommend being restrictive as to who has access to this bootstrap instance. Ideally, a deployment or admin role needs to have access to the bootstrap instance, as the bootstrap instance deploys resources into the given account.

   Example policy:

   ```
   {
       "Version": "2012-10-17",
       "Statement": [
           {
               "Effect": "Allow",
               "Action": "ssm:StartSession",
               "Resource": [
                   "arn:aws:ec2:us-west-2:12345678912:instance/<instance-id>",
                   "arn:aws:ssm:us-west-2:12345678912:document/BootstrapShellDoc-
   <stage>-<region>"
               ]
           }
       ]
   }
   ```

2. To retrieve the instance id of the bootstrap instance that was deployed, run the following command:

   > **Note**
   >
   > Alternatively, you can retrieve the instance id from the Amazon EC2 console.

```
instance=$(aws ec2 describe-instances --filters 'Name=tag:Name,Values=bootstrap-*' --
query 'Reservations[].Instances[].[InstanceId]' --output text)
```

3. Using the instance id obtained from the previous step, run the following command to access the bootstrap instance:

> **Note**
>
> Update the **Stage** if it isn't `dev`, and update the Region if it isn't `us-east-1`.

```
aws ssm start-session --document-name SSM-dev-BootstrapShellDoc-dev-us-east-1 --
target $instance --region us-east-1
```

4. To prepare the bootstrap instance for deploying the migration pieces, run:

> **Note**
>
> The initial setup can take approximately 10-15 minutes.

```
./initBootstrap.sh && cd deployment/cdk/opensearch-service-migration
```

Step 3: Customize the migration options

1. From the same shell on the bootstrap instance, find the `cdk.context.json` file in the current, `/opensearch-migrations/deployment/cdk/opensearch-service-migration`, directory. The `cdk.context.json` file contains a `default` context block that has already been populated with the given stage and network defined in the CloudFormation Parameters. The `targetCluster` and `sourceCluster` details should be filled in before moving to the next step.

   Because multiple migration options might change frequently, refer to Configuration Options for configuration options based on your use case.

> **Note**
>
> The Migration tooling expects the source cluster, target cluster, and migration resources to all exist in the same VPC. If this isn't the case, manual networking setup outside of this documentation is likely required.

Step 4: Deploy the migration stacks

Follow the step-by-step instructions in this section to deploy the migration stacks into your account.

Time to deploy: Approximately 45-60 minutes

1. Specify the Region in which you want to deploy the migration stacks by setting the **AWS_DEFAULT_REGION** environment variable, for example:

```
export AWS_DEFAULT_REGION=us-east-1
```

2. If this is the first time you're deploying CDK in this Region of your account, you must bootstrap the account to work with CDK. In the `cdk.context.json` file, run the following command:

```
cdk bootstrap --c contextId=default
```

3. If this is the first time you're deploying Amazon OpenSearch Service or an ECS cluster with CDK in this account, create the service linked role initially by running:

```
aws iam create-service-linked-role --aws-service-name opensearchservice.amazonaws.com
```

```
aws iam create-service-linked-role --aws-service-name ecs.amazonaws.com
```

4. To deploy the CDK context block you have configured, run the CDK command:

```
cdk deploy "*" --c contextId=default --require-approval never --concurrency 3
```

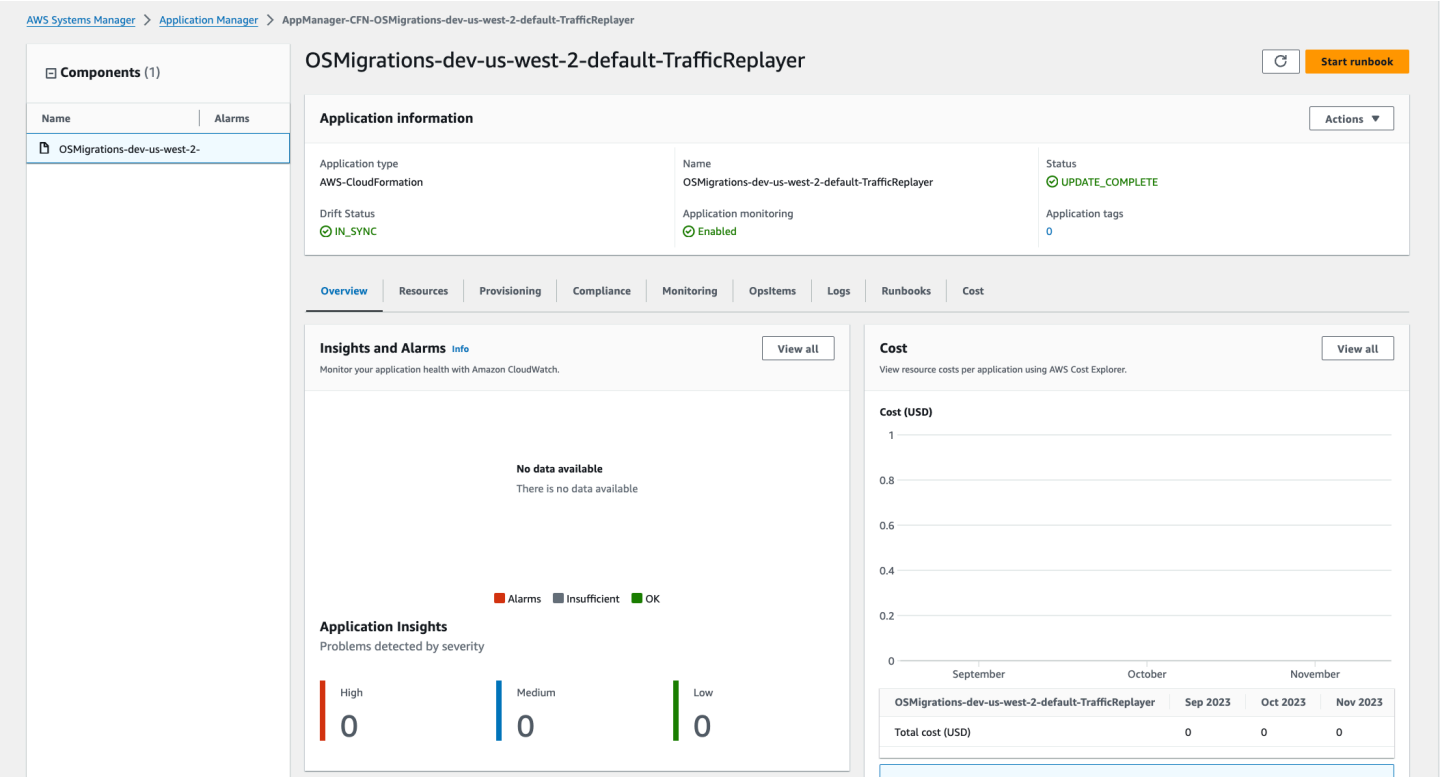
For additional help and tips, see the README.md file in the GitHub repository.

Monitor the solution with Service Catalog AppRegistry

The solution includes a Service Catalog AppRegistry resource to register the CloudFormation template and underlying resources as an application in both Service Catalog AppRegistry and AWS Systems Manager Application Manager.

AWS Systems Manager Application Manager gives you an application-level view into this solution and its resources so that you can:

- Monitor its resources, costs for the deployed resources across stacks and AWS accounts, and logs associated with this solution from a central location.
- View operations data for the solution's AWS resources (such as deployment status, Amazon CloudWatch alarms, resource configurations, and operational issues).



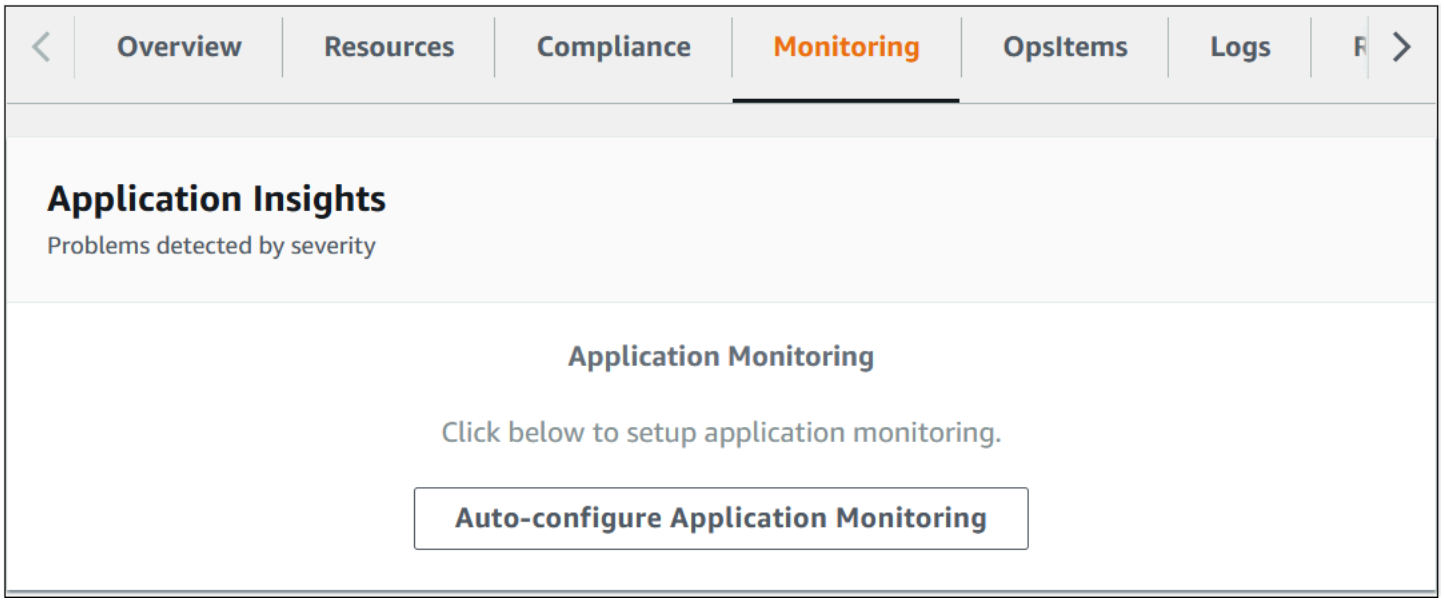
Migration Assistant Traffic Replayer stack in Application Manager

Activate CloudWatch Application Insights

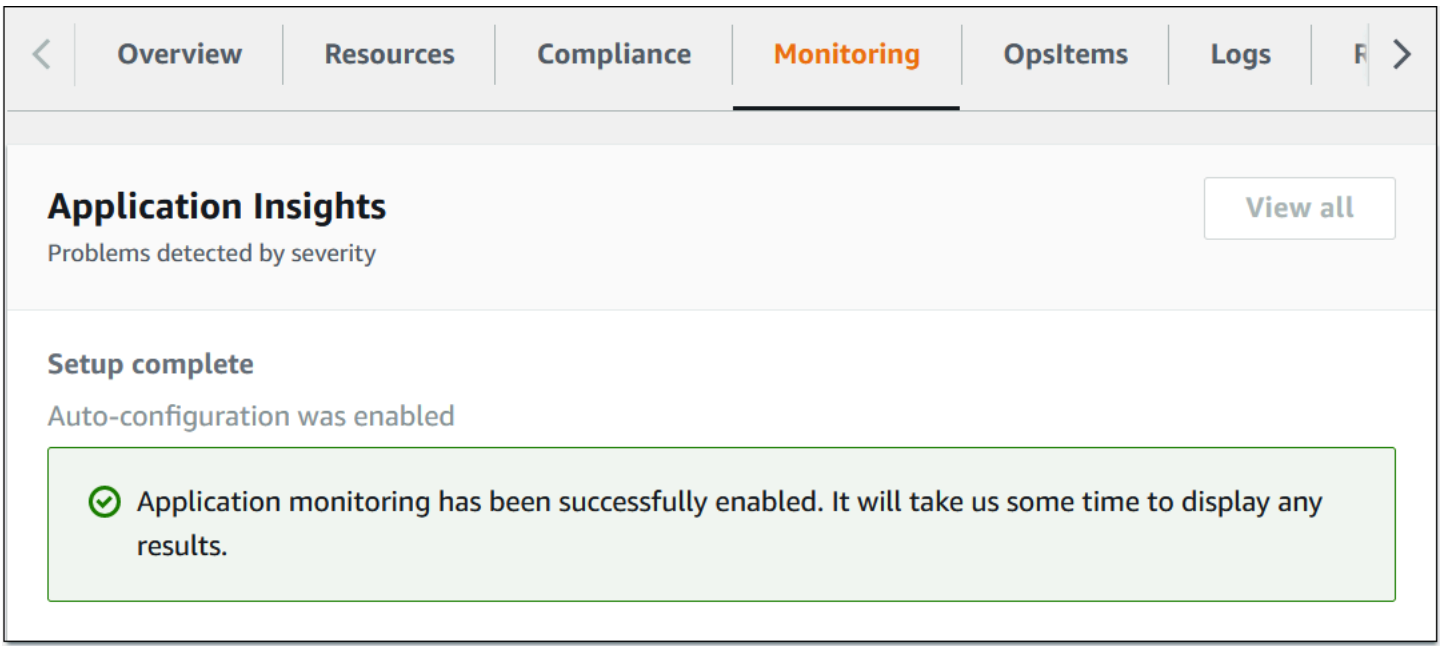
1. Sign in to the Systems Manager console.

2. In the navigation pane, choose **Application Manager**.
3. In **Applications**, choose **AppRegistry applications**.
4. In **AppRegistry applications**, search for the application name for this solution and select it.

   The application name will have App Registry in the Application Source column, and will have a combination of the solution name, Region, account ID, or stack name.
5. In the **Components** tree, choose the application stack you want to activate.
6. In the **Monitoring** tab, in **Application Insights**, select **Auto-configure Application Monitoring**.



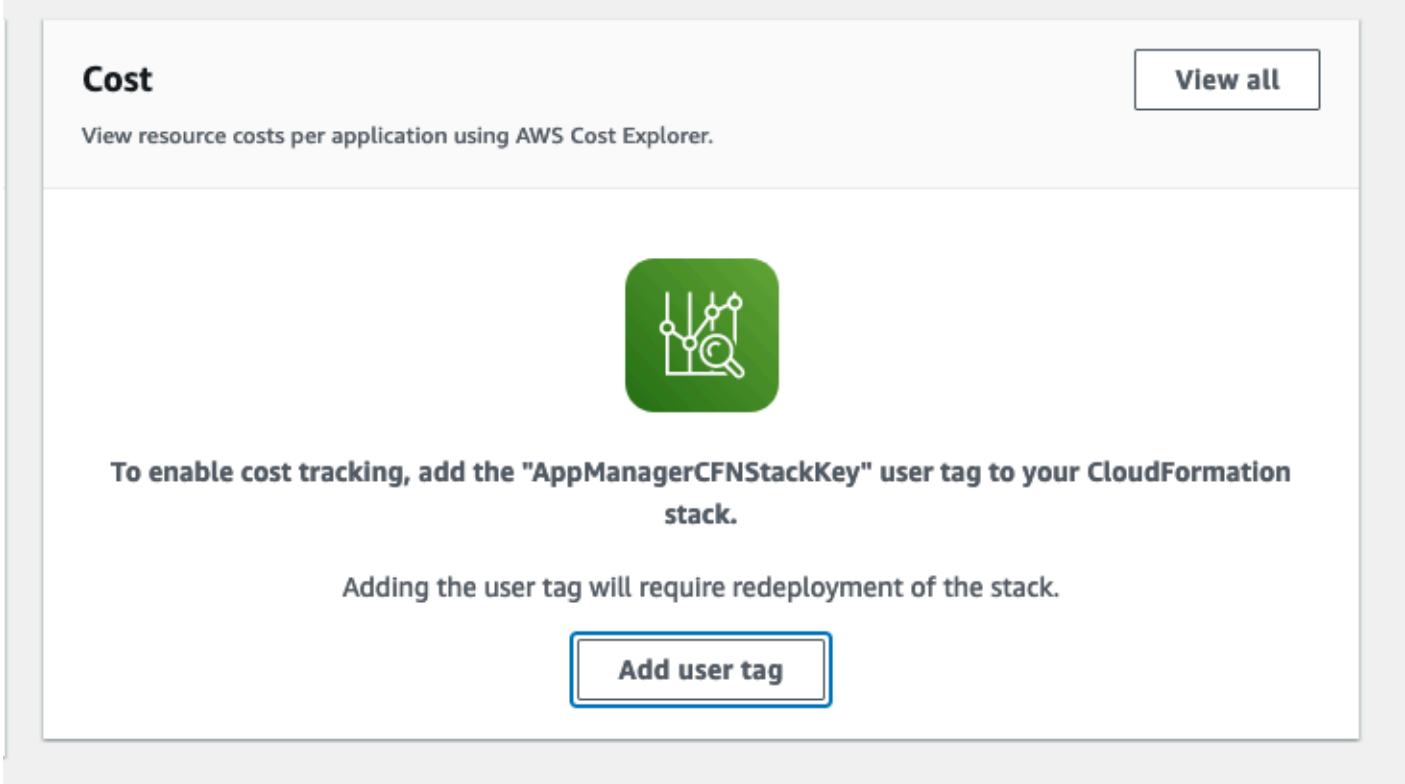
Monitoring for your applications is now activated and the following status box appears:



Confirm cost tags associated with the solution

After you activate cost allocation tags associated with the solution, you must confirm the cost allocation tags to see the costs for this solution. To confirm cost allocation tags:

1. Sign in to the Systems Manager console.
2. In the navigation pane, choose **Application Manager**.
3. In **Applications**, choose the application name for this solution and select it.
4. In the **Overview** tab, in **Cost**, select **Add user tag**.



5. On the **Add user tag** page, enter `confirm`, then select **Add user tag**.

The activation process can take up to 24 hours to complete and the tag data to appear.

Activate cost allocation tags associated with the solution

After you activate Cost Explorer, you must activate the cost allocation tags associated with this solution to see the costs for this solution. The cost allocation tags can only be activated from the management account for the organization. To activate cost allocation tags:

1. Sign in to the AWS Billing and Cost Management and Cost Management console.
2. In the navigation pane, select **Cost Allocation Tags**.
3. On the **Cost allocation tags** page, filter for the `AppManagerCFNStackKey` tag, then select the tag from the results shown.
4. Choose **Activate**.

The activation process can take up to 24 hours to complete and the tag data to appear.

Activate AWS Cost Explorer

You can see the overview of the costs associated with the application and application components within the Application Manager console through integration with AWS Cost Explorer. Cost Explorer helps you manage costs by providing a view of your AWS resource costs and usage over time.

1. Sign in to the AWS Cost Management console.
2. In the navigation pane, select **Cost Explorer** to view the solution's costs and usage over time.

Troubleshooting

This section provides known issue resolution when deploying the solution. If these instructions don't address your issue, see the Contact AWS Support section for instructions on opening an AWS Support case for this solution.

Problem: Requests against the target cluster are failing

If the analytics dashboard shows that requests (that succeeded against the source cluster) are failing when replayed against the target cluster, this might be caused due to several issues.

Resolution

First, check the specific HTTP codes being returned by looking at the status code metrics in the analytics cluster.

1. 403 error - A 403 error is "Unauthorized" and indicates that something is wrong with the configuration—either on the target cluster side or the replayer side with how authorization is configured. Ensure that the authorization strategy selected for each side matches (for example, basic auth vs. SigV4). For basic auth, ensure that the correct auth credentials are provided via the command-line argument or Secrets Manager. If the target cluster does not have any form of authorization enabled, but the source cluster does, ensure that the `--remove-auth-header` flag is provided. For more information about authorization header related flags, see Authorization header for Replayer requests.

   To verify that authorization headers are being applied as expected, check the tuples exported to the EFS volume to see the original and replayed request and responses. To view the tuples, refer to Understanding data from the Replayer.
2. 404 errors - 404 errors generally occur when a document that's being queried or modified can't be found. If you haven't done a historic backfill via snapshot & restore, or the backfill is still in progress, it's possible that a query that returned results on the source cluster will not find the same documents on the target cluster because they haven't been backfilled yet. The resolution here depends on your use case. If you don't intend to backfill and your queries rely on recent documents, the issue will reduce in frequency over time. If you have a planned backfill or if it's already in progress, this will likely clear up when that has completed and all documents are present.

If none of these cases apply and the documents should be present, this might be either due to issues with the backfill or when the document was being replayed against the cluster. This needs to be debugged on a case-to-case basis, but as a good starting point, you can search through the tuples for the document ID in question to check whether it was replayed and successful. Refer to Understanding data from the replayer for more information.

Contact AWS Support

If you have AWS Developer Support, AWS Business Support, or AWS Enterprise Support, you can use the Support Center to get expert assistance with this solution. The following sections provide instructions.

Create case

1. Sign in to Support Center.
2. Choose **Create case**.

How can we help?

1. Choose **Technical**.
2. For **Service**, select **Solutions**.
3. For **Category**, select **Other Solutions**.
4. For **Severity**, select the option that best matches your use case.
5. When you enter the **Service**, **Category**, and **Severity**, the interface populates links to common troubleshooting questions. If you can't resolve your question with these links, choose **Next step: Additional information**.

Additional information

1. For **Subject**, enter text summarizing your question or issue.
2. For **Description**, describe the issue in detail.
3. Choose **Attach files**.
4. Attach the information that AWS Support needs to process the request.

Help us resolve your case faster

1. Enter the requested information.
2. Choose **Next step: Solve now or contact us**.

Solve now or contact us

1. Review the **Solve now** solutions.
2. If you can't resolve your issue with these solutions, choose **Contact us**, enter the requested information, and choose **Submit**.

Uninstall the solution

You can uninstall the Migration Assistant for Amazon OpenSearch Service solution from the AWS Management Console or by using the AWS Command Line Interface. Manually remove the contents of the bucket that matches cdk-*<unique id>*-assets-*<account id>*-*<region>* created by this solution. Migration Assistant for Amazon OpenSearch Service does not automatically delete S3 buckets in case you have stored data to retain.

> **Important**
>
> After your migration is complete, remove all solution resources.

Using the AWS Management Console

1. Sign in to the CloudFormation console.
2. On the **Stacks** page, select this solution's installation stack. Depending on what options you enabled for the deployment, the solution creates different AWS CloudFormation stacks. The stacks must be deleted in the following order:
   a. `MigrationConsole`
   b. `TrafficReplayer`
   c. `CaptureProxy`
   d. `CaptureProxyES` (only available for demo install)
   e. `ReindexFromSnapshot`
   f. `MigrationInfra`
   g. `NetworkInfra`
3. On the **Stacks** page, delete the bootstrap stack.
4. Choose **Delete** for each of the previous.

Using AWS Command Line Interface

To remove all the CDK stacks, which the solution creates during a deployment, you can run a command similar to the following within the CDK directory.

Log in to the `bootstrap` container, as performed in Set up the Bootstrap Instance, and run the following commands:

```
cd deployment/cdk/opensearch-service-migration
  $ cdk destroy "*" --c contextId=default
```

After the Migration Assistant for Amazon OpenSearch Service solution stacks *(see step 2. in the Using the AWS Management Console section)* have been removed, remove the bootstrap CloudFormation template.

Determine whether the AWS Command Line Interface (AWS CLI) is available in your environment. For installation instructions, see What Is the AWS Command Line Interface in the *AWS CLI User Guide*. After confirming that the AWS CLI is available, set the default Region to the Region that was used for deployment. Then perform the AWS CLI command to delete the given AWS CloudFormation bootstrap stack name:

```
export AWS_DEFAULT_REGION=<aws_region>
  $ aws cloudformation delete-stack --stack-name <bootstrap_stack_name>
```

Use the solution

For information on how to use this solution, refer to the Migration Assistant for Amazon OpenSearch Service wiki.

Developer guide

This section provides the source code for the solution.

Source code

Visit our GitHub repository to download the source files for this solution and to share your customizations with others.

> **Note**
>
> Report technical issues with the solution and feature requests on the Issues page of the GitHub repository.

AWS CDK generates the solution's bootstrap template. See the README.md file for additional information.

After the Migration Assistant for Amazon OpenSearch Service bootstrap is installed, install additional resources using CDK templates. For more information, refer to the opensearch-migrations GitHub repository.

Reference

This section includes information about an optional feature for collecting unique metrics for this solution and a list of builders who contributed to this solution.

Anonymized data collection

This solution includes an option to send anonymized operational metrics to AWS. We use this data to better understand how customers use this solution and related services and products. When invoked, the following information is collected and sent to AWS:

- **Solution ID** - The AWS solution identifier
- **Unique ID (UUID)** - Randomly generated, unique identifier for each Migration Assistant for Amazon OpenSearch Service deployment
- **Timestamp** - Data-collection timestamp

AWS owns the data gathered though this survey. Data collection is subject to the Privacy Notice. To opt out of this feature, complete the following steps before launching the AWS CloudFormation template.

1. Download the AWS CloudFormation template to your local hard drive.
2. Open the AWS CloudFormation template with a text editor.
3. Modify the AWS CloudFormation template mapping section from:

```
AnonymizedData:
    SendAnonymizedData:
      Data: Yes
```

   to:

```
AnonymizedData:
    SendAnonymizedData:
      Data: No
```

4. Sign in to the AWS CloudFormation console.
5. Select **Create stack**.

6. On the **Create stack** page, specify template section, then select **Upload a template file**.
7. Under **Upload a template file**, choose **Choose file** and select the edited template from your local drive.
8. Choose **Next** and follow the steps in Launch the stack in the Deploy the solution section of this guide.

Contributors

- Kartik Ganesh
- Chris Helma
- Omar Khasawneh
- Tanner Lewis
- Brian Presley
- Greg Schohn
- Himanshu Setia
- Mikayla Thompson
- Hang (Arthur) Zuo

Revisions

Date	Change
November 2023	Initial release
September 2024	Release v2.0 • Added details about Reindex-From-Snaps hot for Backfill • Updated cost structure and provided estimates for different use cases (for example, metadata migration, backfill, live capture) • Revised capture traffic design for zero-down time migration support • Added increased limits because we now support 100 TB migrations with higher network traffic throughput
December 2024	Release v2.1.0: Added deployment options for Virtual Private Cloud, support for Elasticserach 6.8 as a data source, support for customizable data/metadata transformations, and improved performance for backfill. For more informati on, refer to the CHANGELOG.md file in the GitHub repository.

Notices

Customers are responsible for making their own independent assessment of the information in this document. This document: (a) is for informational purposes only, (b) represents AWS current product offerings and practices, which are subject to change without notice, and (c) does not create any commitments or assurances from AWS and its affiliates, suppliers, or licensors. AWS products or services are provided "as is" without warranties, representations, or conditions of any kind, whether express or implied. AWS responsibilities and liabilities to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.