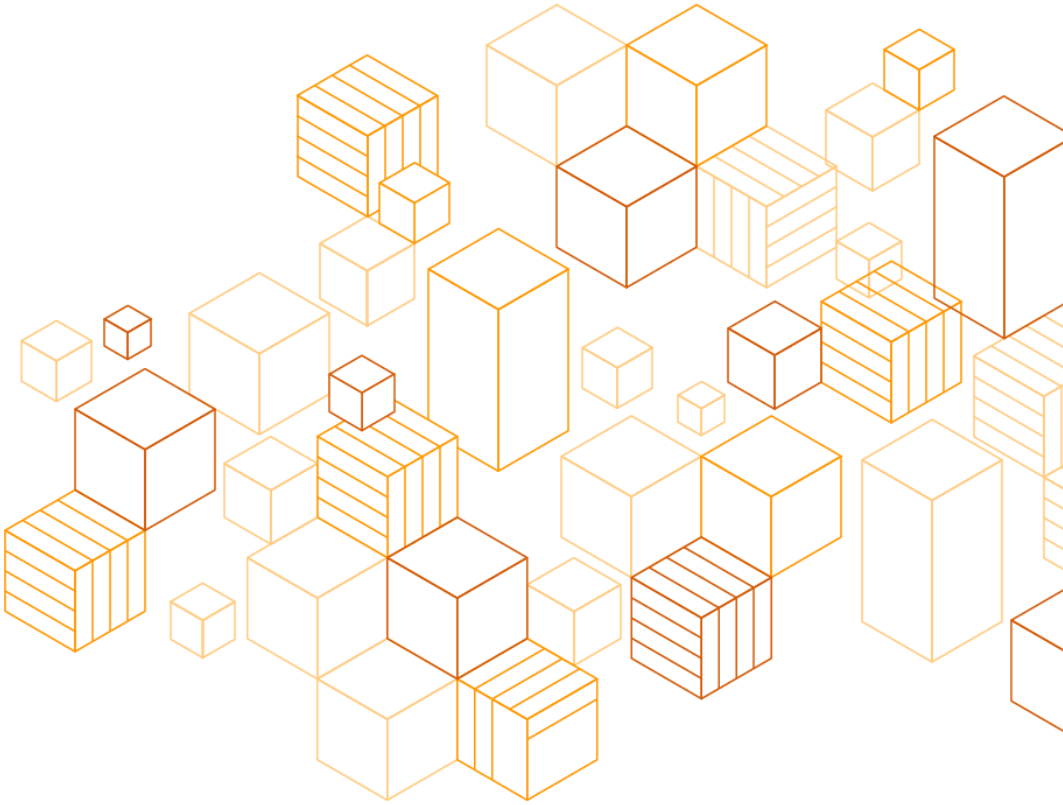
Amazon OpenSearch Service Migration Guide

Technical Guide

January 24, 2023



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 current AWS 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. The responsibilities and liabilities of AWS 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.

Contents

About this Guide

For many customers, migrating self-managed Elasticsearch or OpenSearch deployments to Amazon OpenSearch Service is challenging. The common challenges are of workload assessment, capacity planning, and architectural optimization. Furthermore, how to meet all requirements of operational analytics applications from on-premises data centers to an AWS cloud raises extra questions. This guide covers the overall journey of a migration to Amazon OpenSearch Service, and provides best practices that AWS experts have accumulated over time. The step-by-step instructions help customers conduct their migrations in an effective and efficient approach. Note that this guide mainly covers Amazon OpenSearch Service provisioned domains and not Amazon OpenSearch Serverless collections.

Overview

OpenSearch is a distributed, open-source search and analytics suite used for a broad set of operational analytics use cases like real-time application monitoring, log analytics, data observability, and application and product catalog search. OpenSearch provides capabilities for providing low latency search response, and fast access to large volumes of data with an integrated open-source data visualization tool called *OpenSearch Dashboards*.

Amazon OpenSearch Service makes it easy for you to perform interactive log analytics, real-time application monitoring, website search, and more. Amazon OpenSearch Service offers the latest versions of OpenSearch, support for 19 versions of Elasticsearch (1.5 to 7.10 versions), as well as visualization capabilities powered by OpenSearch Dashboards and Kibana (1.5 to 7.10 versions). Amazon OpenSearch Service currently has tens of thousands of active customers with hundreds of thousands of clusters under management processing hundreds of trillions of requests per month.

Managing OpenSearch or Elasticsearch clusters on-premises or on cloud infrastructure is highly complex, expensive, and tedious work. You must provision and maintain the infrastructure to run these clusters. The efforts include 1) hardware procurement and set up, 2) software installation, configuration, patching, and upgrading, 3) reliability and availability considerations, 4) performance and scalability considerations, and 5) security & compliance considerations, such as network isolation, fine-grained access control, encryptions, and compliance programs such as PCI, HIPAA, FedRAMP, ISO, GDPR, and SOC.

In this guide, you will learn the approaches and best practices of migrating on-premises, or self-managed open-source Elasticsearch or OpenSearch to the fully managed Amazon OpenSearch Service.

Benefits of Migrating to Amazon OpenSearch Service

Easy to deploy, and manage

Amazon OpenSearch Service simplifies management tasks like hardware provisioning, software installation and patching, failure recovery, backups, and monitoring. You don't need to have a dedicated team of OpenSearch experts to manage your clusters. An OpenSearch cluster in Amazon OpenSearch Service is also called domain, we will use cluster and domain terminology interchangeably in this document. Amazon OpenSearch Service provides cluster health monitoring through Amazon CloudWatch service, you can setup alerts so you can get notified of any changes to your clusters' health. AWS support is one-on-one, fast-response technical support from experienced engineers. Customers with operational challenges or technical questions can contact AWS support and receive personalized support with reliable response times.

Cost-effective

Amazon OpenSearch Service provides a full array of advanced capabilities without charging additional licensing fees. You can leverage capabilities such as enterprise-grade security, real-time alerting, cross-cluster search, automated index management, and anomaly detection at no extra cost. There are no inter-Availability Zone (inter-AZ) data transfer charges and hourly snapshots are provided at no extra cost. With *UltraWarm*, you can run fast, interactive analytics on up to three petabytes of log data while reducing the cost per GB by up to 90% compared to the hot storage tier. Furthermore, Amazon OpenSearch Service offers reserved instances (RIs) that provide significant discounts compared to the standard on-demand instances so you can build log analytics and search solutions at the scale and speed you need. For more information please visit - https://aws.amazon.com/opensearch-service/features/cost/

More scalable and reliable

Amazon OpenSearch Service enables you to store petabytes of data in a single cluster. You can easily query data across multiple clusters, which enables you to analyze all of your data via a single OpenSearch Dashboards interface. It is designed to be highly reliable using multi-Availability Zone (AZ) deployments so that you can replicate data between up to three AZs in the same region. There is zero downtime when you make software updates and upgrades or scale your environment.

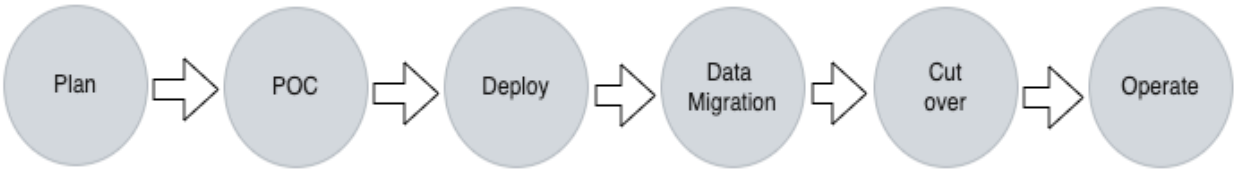
Secure and Compliant

Amazon OpenSearch Service takes care of all security patches and offers network isolation via VPC, fine-grained access control, and multi-tenant OpenSearch Dashboards support. You can choose to encrypt your data at rest and in transit. Amazon OpenSearch Service is also HIPAA eligible and compliant with PCI DSS, SOC, ISO, GDPR and FedRAMP standards to help you meet industry-specific and regulatory requirements. For more information visit https://aws.amazon.com/opensearch-service/features/security/

Migration Journey

Depending up on your current deployment, migrating to an Amazon OpenSearch Service can be a simple or a complex procedure involving multiple steps like right sizing cluster, version upgrade, re-indexing, storage tiering, determining number of shards and more. In this section, you will explore a migration approaches and key considerations in each step of the process. We will also discuss the best practices based upon AWS's experience of helping migrate many AWS customers from existing tooling to Amazon OpenSearch Service. In addition, we will discuss what constitute an effective migration strategy.

The following diagram depicts a typical migration journey. You may be migrating from a self-managed Elasticsearch or OpenSearch cluster or you may be migrating from another technology to Amazon OpenSearch Service. In most cases, the steps will remain the same. The time you spend on each step will vary based on the complexity of your environment. We start with a careful planning activity, followed by running a proof of concept to ensure that the target environment meets your cost, security, performance, and migration goals. The POC activity is followed by deploying the target environment and moving the data to it. You ensure your data is synchronized between the current environment and the new environment after which you can execute a cut over to the new environment. Once you have successfully cutover you will operate the environment following operational best practices. We will go through each stage in detail in the following sections.



Stage 1 - Planning

Migration starts with planning for a target environment that you are going to build to meet your requirements. Planning involves looking at a set of focus areas, each of which will require careful consideration. These focus areas will help you achieve your migration goals by reducing the migration complexity and costs, and will help you make decisions that will form the migration strategy. During the planning stage it is critical to assess your current environment and identify pain points that you want to address as part of this migration. These pain points can be around performance, security, reliability, speed of delivery, cost or ease of operations. Following diagram depicts the focus areas

Key Focus Areas during planning

Figure 1 Key focus areas of planning

Sizing

Sizing helps you determine the right instance type, number of data nodes, and storage requirement for your target environment. We recommend that you go by the rule of thumb, size first by the storage then CPU. Note that if you're already a user for Elasticsearch or OpenSearch, the sizing will generally remain the same. However, you will need to identify the right instance type equivalent to your current environment. We recommend you follow the instructions below to help determine the right size.

Storage

Sizing your cluster starts with defining the storage requirements. Identify the raw storage that you need for your cluster. This is determined by assessing data generated by your source system (e.g. servers generating logs, or product catalog raw size). After identifying how much raw data you have, you can use the formula given below as a rule of thumb. This will ensure that you have sufficient storage. Use the calculated number as a starting point for your proof of concept.

storage needed = (daily source data in bytes * 1.45) * (number_of_replicas + 1) * number of days retention.

We will explain the formula now. The on-disk size of an index varies, but is often 10% larger than the source data. There is operating system overhead of 5% that is reserved by Linux for system recovery and to safeguard against disk defragmentation problems. OpenSearch Service also reserves 20% of the storage space of each instance for segment merges, logs and other internal operations. We also recommend keeping 10% additional storage to help minimize the impact of node failure and Availability Zone outages. Combining all these overheads and reservations require 45% additional space than the actual raw data in the source. That's why you see we multiple source data by 1.45 in above calculation. We multiply this by number of copies of data (i.e. 1 primary + n replica) and number of days you want to retain data in hot storage tier. Replica count depends on your resiliency and throughput requirement. For a average use case, you start with 1 primary and 1 replica.

OpenSearch offers hot, warm and cold storage tiers. Warm storage tier is provided using UltraWarm storage. UltraWarm provides a cost-effective way to store large amounts of read-only data on Amazon OpenSearch Service. Standard data nodes use "hot" storage, which takes the form of instance stores or Amazon EBS volumes attached to each node. Hot storage provides the fastest possible performance

for indexing and searching new data. On the other hand, UltraWarm nodes use Amazon S3 and a sophisticated caching solution to improve performance. For indices that you are not actively writing to, or query less frequently, and do not have the same performance requirements, UltraWarm offers significantly lower costs per GiB of data. More information on UltraWarm can be obtained from AWS docs.

For hot storage, when we create OpenSearch domain, depending upon the choice of instance type of data nodes we may have to define EBS volume size. You would calculate storage requirement using formula given above to determine the volume size for EBS backed instances. We recommend using gp3 volumes for latest generation T3, R5, R6G, M5, M5g, C5, and C6g instance families. Amazon EBS gp3 enables customers to provision performance independent of storage capacity, provides better baseline performance, at a 9.6% lower price point per GB than existing gp2 volumes on OpenSearch Service. In addition, with gp3 you now get denser storage on R5, R6g, M5, M6g instance families, which can help you to further optimize your costs. You can create EBS volume up to the supported limit. For more information on limits please see Amazon OpenSearch Service Limits.

For data nodes that have NVME drives such as i3 and r6gd instances, the volume size is fixed and the EBS volumes are not an option. NVME is fastest local storage available, it offers superior IO performance than EBS storage.

Number of nodes and Instance types

Number of nodes is determined based on the number of CPU required to operate your workload. Number of CPU is determined based on the shard count. An index in OpenSearch is made up multiple shards. When you create index, you decide number of shards for the index. Therefore, you will need to calculate total shard count you intend to store in the domain first, then determine the CPU, and then find the most cost-effective node type and count that gives you required number of CPU and storage. This is usually a starting point, you will run your tests to determine that the estimate size is meeting your functional and non-functional requirements.

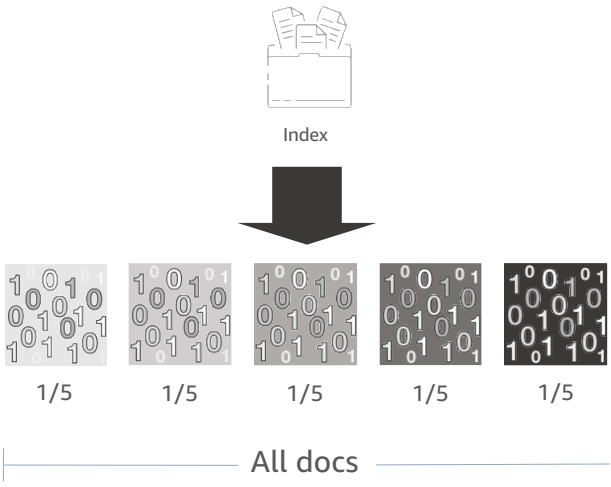


Figure 2 Index is composed of shards

Following is how you determine the shard count.

Shard Count:

Once you know the storage requirements, you can identify the shard count based on guideline give below. identifying the number of shards will help in finding out the right instance types that suites your workload. The shards shouldn't be too large or too numerous. Large shards can make it difficult for OpenSearch to recover from failure, but because each shard uses some amount of CPU and memory, having too many small shards can cause performance issues and out of memory errors.

- Search use cases – 10 to 30 GB / shard
- Log Analytics – 50 GB / shard

Generally, search use cases would have 1 or few indices each representing a searchable entity or a catalog. For log analytics use case, an index could represent daily or weekly log file. Once you have decided how many indices, begin with the above scale guidance and you will derive appropriate shard size after a POC. Imbalance in shard allocation can lead to skewing. Note that you would need to ensure your shards are evenly distributed over the data nodes and there is no storage or processing skew. Balancing shard count evenly across AZ also helps improve stability.

CPU

Next step is identifying how much CPU you would need for your workload. Based on our experience with many AWS customers, we recommend starting with CPU count 1.5 times that of your active shards. An active shard is any shard for an index that is receiving substantial writes. Use the primary shard count to determine active shards for indices that are receiving substantial writes. For log analytics, only the current index is active. For search use cases, which are read heavy, use the primary shard count. Although 1.5 is recommended, this is highly workload-dependent. Be sure to test and monitor CPU utilization and scale accordingly.

Best practice to maintain your CPU utilization is to make sure that OpenSearch Service has enough resources to perform its tasks. A cluster that has consistently high CPU utilization can degrade cluster performance. When your cluster is overloaded, OpenSearch Service will block incoming requests resulting in request rejections. This is to protect domain from failing. General guidelines on the CPU usage will be about 60% average, 80% max CPU utilization with occasional spikes of 100% are still acceptable and may not need scaling or reconfiguration.

Instance Types

Amazon OpenSearch Service provides you options with several instance types. Depending on the use case, you can arrive at the best fit instance types for your Amazon OpenSearch Service domain. Based on various testing that was performed, for Data Nodes, a general guideline is to use graviton instances for log analytics use cases because these use cases may need a lot of JVM because these are write heavy use cases with much lesser percentage of reads. For search use cases, the recommendation is to use R and C series graviton instances for data nodes since it requires more CPU compared to log analytics use cases. M for smaller workloads for both search and logs. I series offer NVME drives and are used by customers with fast indexing and low latency search requirements.

Amazon OpenSearch supports different instance classes: R, C, M, T, I. You choose an instance class based on the workload i.e., memory intensive, compute intensive or mixed. For the chosen instance class, choose the latest generation instance type. Graviton and later generations are generally recommended node types as they are built-in to provide better price/performance improvement over previous generation instances

Cluster will need Cluster manager (previously called Leader) nodes and data nodes. It's a good practice to pick different instance types for dedicated cluster manager nodes and data nodes. Cluster manager nodes manage the state of the cluster which needs more CPU and less JVM. The preferred option is to use C5 instances for cluster manager nodes unless you have a very large cluster where you may want to look at R series. If you are observing consistent CPU spikes, scale up your cluster or move to C series.

AWS offers general purpose (M6g), compute optimized (C6g), and memory optimized (R6g, R6gd) for Amazon OpenSearch Service version 7.9 or later powered by AWS Graviton2 processors. These instances are built using custom silicon designed by Amazon. These instances are Amazon-designed hardware and software innovations that enable the delivery of efficient, flexible, and secure cloud services with isolated multi-tenancy, private networking, and fast local storage

Graviton2 instance family provides up to 50% reduction in indexing latency, and up to 30% improvement in query performance when compared to the current generation (M5, C5, R5).

Functionality

Functionality focus area helps you ensure that you do not lose any functionality when you migrate to a target Amazon OpenSearch Service environment. We will pay close attention to following aspects

Current solution functionality: We recommend that you analyze your current solution and determine the features, plugins, APIs that you use in the current technology stack (e.g. Elasticsearch, OpenSearch, other solution). You may also determine what functionality is critical to your business, what can be modified, and what can dropped during the migration.

Amazon OpenSearch Service functionality: We recommend that you perform an analysis of the latest OpenSearch version, features it offers, and plugins that are available in Amazon OpenSearch Service to ensure functionality that is required is available once migrated. You want to determine that the target platform supports the features you need e.g. index state management, which automates rolling over of the indices or machine learning features such as anomaly detection. You would also perform mapping of existing functionality of your current solution to features in Amazon OpenSearch Service that provide you equivalent capability so you can continue to support your workloads.

You can find information about the functionality available within each supported version of the Elasticsearch or OpenSearch software in Amazon OpenSearch Service documentation.

Packaged Plugins: Amazon OpenSearch Service supports a number of plugins that are part of opensource OpenSearch project. If you are using any *licensed plugin* from Elasticsearch suite that is part of X-Pack or otherwise, you may want to determine an equivalent plugin or native feature within OpenSearch ecosystem. OpenSearch has a number of plugins that provide enterprise-grade features equivalent to those licensed plugins. You may also want to capture that as a point to prove in the POC stage. While Amazon OpenSearch Service supports a number of OpenSearch plugins out of the box, however, you may be using an OpenSearch plugin that may not be available within Amazon OpenSearch Service at the moment. Please contact AWS to request adding the plugin to the Amazon OpenSearch Service future roadmap. OpenSearch Service documentation contains a list of plugins by versions which can be a good place to determine the correct plugin and version in the target environment.

Custom Plugins: If you have customizations, such as custom plugins to support your solution, you may need to carefully analyze the functionality to see if you will be able to port it to the target environment using Amazon OpenSearch Service supported plugins or to native features within OpenSearch. At the time of writing this document, custom plugins are not supported, therefore you will need to consider

alternate ways to deliver the custom plugin function and experience. It is recommended that all such decisions are proven during the POC stage. Migration is a good time to evaluate current solution functionality to see if it is critical to your business.

Version Dependencies

Version dependencies focus area helps you build roadmap of your migration journey through various versions to reach the latest version of Amazon OpenSearch Service. Following are key points to consider.

Selecting engine version: It is very important to consider version dependencies carefully. Amazon OpenSearch Service supports a number of Elasticsearch versions and all major OpenSearch engine versions (*latest version of OpenSearch can take few weeks to be supported in Amazon OpenSearch Service from the date of release*). We recommend that you go through the feature supported by engine version in Amazon OpenSearch Service documentation to make the right version selection. Identifying compatible version makes it simple and easier to execute a migration. For example: if you are currently running version 5.x of Elasticsearch, an equivalent engine version will be available in the OpenSearch Service (5.x Elasticsearch). Choosing the same major (and closest minor) version allows you to use the snapshot restore approach to migrate. This is often the simplest approach in many cases.

Upgrade to the latest version - While you may be able to operate equivalent older version of software in Amazon OpenSearch Service, we highly recommend upgrading to the latest available version of the Amazon OpenSearch Service software. This helps you take advantage of the performance improvements, reliability, cost savings, and many new features that are available in the latest versions of the engine. Migration is a good opportunity to eliminate the technical debt that may exists due to running older versions of software.

Version Upgrade Strategy: If you decided that you want to upgrade to the latest version of the software during the migration, you will need to determine the steps and an upgrade strategy. Amazon OpenSearch Service documentation provides information on upgrade paths. It is important to understand the breaking changes between different versions which may (in some cases) require you to plan for adjustments to your index modeling and design.

***Note on multiple mapping types (doctypes) breaking change*:** *Multiple mappings* functionality is available only in Elasticsearch versions 5.x and lower. Indices created in versions 6.x and above only support a single mapping type per index. If you are using multiple mapping types, it is recommended to remodel that data into multiple indices.

In case of a time-sensitive migration, consider a simple option where you perform an equivalent version migration (example: 5.x to 5.x), and then upgrade to OpenSearch at a later date. OpenSearch Service offers in-place upgrades for domains that run Elasticsearch versions 5.1 (if compatible) or newer, and OpenSearch 1.0 or newer. You should perform a test to see if your indices are compatible for in-place upgrades when you are running Elasticsearch version 5.x. This means you may be able to migrate to equivalent version, and then perform an in-place upgrade once you have made the necessary changes to make your indices and other functionality compatible with the latest version. You should review the upgrade domain documentation carefully.

Pre-upgrade checks: Amazon OpenSearch Service upgrade functionality can perform *pre-upgrade checks* by scanning the environment to determine issues that can block the upgrade. It is recommended that you address the reported issues before proceeding with the upgrade.

KPIs and Business Continuity

It is essential that during the migration you establish your business goals and key performance indicators (KPIs) to measure success. It is important to determine your goals at the beginning of the migration process. Some of the common goals we see in our customer journeys are:

- Improve operational agility

Under this goal you can measure and compare existing deployment with the target environment using some of the following metrics.

    - Mean time to provision cluster.
    - Time to rollout the deployment to a new geography
    - Mean time to configure cluster security
    - Mean time to scale your environment (add nodes, add storage etc.)
    - Mean time to detect slow performing query and mean time to repair.
    - Mean time to upgrade software version
- Reduce total cost of ownership (TCO)

Following are some of the metrics you can measure to calculate the TCO. Please note that this list is not exhaustive.

    - No. of staff hours to build and operate the solution (development, devops, monitoring, scale, backup, restore)
    - License cost associated with the existing software
    - Data center costs (hardware procurement/refresh, electricity, cooling, space, racks, networking gears)
    - Staff hours to configure solution (software installations, networking)
    - Cost for compliance audits (HIPAA, PCI DSS, SOC, ISO, GDPR, FedRamp)
    - Cost of configuring security (at rest and in-transit encryption, configuring authn/authz, fine grain access control)
    - Cost of retaining large volume of warm and cold data.
    - Cost of configuring high availability across availability zones
    - Cost of overprovisioning to avoid frequent hardware procurement or handling peak loads.
- Up time & other SLAs

Following are some of the SLAs you can measure and improve with the new environment

    - Total up time (historical uptime data of existing deployment vs 99.9% SLA provided by Amazon OpenSearch Service).
    - Failure recovery (recovery point objective, and recovery time objective).
    - Response time associated with various functions (search, indexing)
    - No. of concurrent users
    - Replication time between different geographies/clusters.

As you migrate to Amazon OpenSearch Service there should be an iterative process to verify whether you are meeting or exceeding those KPIs and whether you achieving the desired outcomes. It is critical to establish a baseline for your current system so that you can determine measurable improvements.

Operational Performance: One of the key areas to look at in your current solution is performance metrics. You should establish a benchmark and determine improvements that you expect to achieve

within your target environment. This includes your **uptime SLA and latency requirements**. This will help you establish and, in most cases, improve your current service levels. Usually customers look at the following service level indicators

- Reads and writes per seconds
- Read and write latency
- Uptime percentage

It is important to fully understand the Amazon OpenSearch Service ("Amazon OpenSearch Service") - Service Level Agreement ("SLA") to architect for your own SLAs.

Process Performance: To establish business continuity goals, it is important to assess your current process performance. Identify and review existing runbooks or SOPs of the current platform and determine areas where your team spends most of its time. Migration is a good opportunity to work on improving those areas so your team can focus on innovating, building business functionality, and improving the customer experience. You can identify pain points of your existing environment by going through the historical support or trouble ticket data to determine time spent by your support and development staff resolving these issues. Capturing the following metrics can help you measure improvements delivered by your target environment:

- Mean time to failure (MTTF) (uptime)
- Mean time between failures (MTBF)
- Mean time to detect (MTTD) a failure
- Mean time to repair (resolve) (MTTR)
- Number of support tickets received, and mean time to resolve those tickets

Seamless transition to new services: To ensure business continuity of your services it is important to carefully plan a seamless transition. Migration is a good time to modernize your application and services associated with your search or log analytics platform. However, you want to plan a careful **cutover** strategy to not impact your existing services. The cutover strategy section in this document provide information on how to plan a seamless cutover to the target environment.

Financial Metrics: There could be many reasons to migrate to OpenSearch Service, but cost is generally a major factor. Understand the total cost of ownership (TCO) of the existing environment so you can measure the cost savings you get by moving to the managed service. You may start with the list of metrics provided above under “reduce total cost of ownership” goal. AWS has published cloud value benchmarking study which can help teams making a business case for migrating to cloud. While the study is not specific to Amazon OpenSearch Service, it provides key value areas which are common across most cloud migrations including migrating to Amazon OpenSearch Service.

In most cases OpenSearch Service delivers lower TCO. When calculating the total cost of ownership, it is critical to incorporate staffing cost. Having understanding of the time and cost your engineers spend to maintain the current environment is an important factor. Many customers just compare the cost of storage, compute, and networking infrastructure with the cost of the managed service. However, that may not provide you with an accurate total cost of ownership. OpenSearch Service provides operational efficiencies to your team by taking away tasks that otherwise had to be performed by your engineers. This includes tasks such as scaling cluster by adding or removing nodes, patching, in-place upgrades, taking backups and configuring monitoring tools to capture logs and metrics. These activities are automated by the service which allows your staff to focus on activities that add direct value to your business whereas AWS offers a production-level support team.

Operations & Security

Operating a self-managed Elasticsearch or OpenSearch cluster can be challenging. You are responsible for provisioning nodes, adding storage, installing and patching operating system, configuring and maintaining high availability, scaling, and many other activities that will be taken care of by the managed service. Therefore, when you migrate to OpenSearch Service, your operational activities will change. Instead of taking care of low-level software configuration and infrastructure provisioning, you will now be able to focus your attention on high-level activities. OpenSearch Service offers logging, monitoring, and troubleshooting features that you will need to become familiar with.

Runbooks and new processes: During the planning stage you will have to identify existing processes that will need to be modified or eliminated. This will allow you to add new operational processes that you may not have had bandwidth for in the past. While OpenSearch Service takes away the undifferentiated heavy lifting, you will still need to ensure your application is designed and monitored to deliver the best performance. You will need to configure monitoring and alerting for your domain so that you are fully aware of any healthy issues due to internal or external factors. You will need to schedule and initiating upgrades to latest versions, etc. All such operational activities will require creating and modify existing runbooks. If you have not been using runbooks, it is a good time to consider doing so. To monitor infrastructure and to analyze operational metrics in Amazon OpenSearch Service it's crucial to maintain runbooks. Runbooks ensure that you operate consistently as per your compliance and regulatory requirements. You should create processes to periodically execute pre-planned steps to ensure remediation processes such as recovery from application crashes and unexpected failures are fully automated.

Support and ticketing system: You may plan and operate a ticketing system to capture incidents associated with your deployments (you may already be doing so). You may need to train your support staff on how to create support tickets with AWS support. You should streamline the process of escalations during ticket triage.

The operational excellence section later in this guide will provide you with a number of best practices and areas that you may need to consider in your runbooks and build processes around.

Security: At AWS, security is the top priority. Amazon OpenSearch Service provides multi-layer security. The service takes care of all security patches and offers network isolation via VPC, fine-grained access control, and multi-tenant support. Your data is encrypted at rest using keys you create and control through AWS Key Management Service (KMS), and the node-to-node encryption capability provides Transport Layer Security (TLS) for all communications between instances in a cluster. Amazon OpenSearch Service is also HIPAA eligible, and compliant with PCI DSS, SOC, ISO, and FedRamp standards to help you meet industry-specific or regulatory requirements.

During planning stage you need to identify the people and processes that interact with the domain, choose a network topology, and plan for authentication and authorization for each principal. Depending upon your organizational security and compliance requirements, you can leverage multiple security features to create an environment that meets your business needs.

VPC: Amazon OpenSearch Service can be configured within AWS VPC, this is the recommended configuration. We do not recommend creating a domain with a public endpoint. Plan to create the necessary network architecture to allow your client applications and users to access the target environment.

Authentication: Amazon OpenSearch Service supports multiple ways to authenticate a user or software client. It supports Amazon Cognito or SAML authentication with your existing identity

provider to access Amazon OpenSearch Dashboard. It also offers integration with IAM identities, and basic HTTP authentication using an internal user database. You should plan to configure and test an appropriate option for authentication. Please refer to security documentation for more detail.

Authorization: We recommend that you follow least privileged access principal in configuring access to the service. Amazon OpenSearch Service provides fine-grained access control to help you configure access at document, row, and column levels.

Familiarize yourself with the security features and test them during the POC stage.

Training

When starting your migration journey to AWS your software development, operations, support, and security teams need to be equipped with knowledge of the OpenSearch Service. You have to consider all teams that interact with your solution. If you are migrating from an Elasticsearch or an OpenSearch environment, most of the knowledge can be carried over. Consider providing training to the following teams:

Software Development team: Your software development team should be educated on the APIs and features such as mechanisms for configuring data ingestion.

Operations team: Your operations team should be trained on how to interact with Amazon OpenSearch domains, how to monitor operational metrics and access logs using CloudWatch. They should learn how to set up automated alarms to warn when OpenSearch Service domains need attention. If you are migrating from an existing toolset that you use on-premises, such as Splunk, what are the monitoring options in Amazon OpenSearch Service that can provide similar visibility into your workloads.

Support team: Your support team should be educated on how to execute runbooks involving OpenSearch Service resources. You may want to update runbooks and event management procedures to make use of AWS support services.

Security team: Your security team should be educated on how to configure fine-grained access control and how to integrate with existing identity providers (IDPs).

Training Options

Training & Certifications - AWS training and certification provides many ways with which you can get started with Amazon OpenSearch Service. We offer both digital and classroom trainings for beginner to professional level on cloud skills required to build, and operate solutions on AWS platform. The content is created by experts at AWS and updated regularly. You have multiple training options.

You can work with your AWS account team to help you identify an appropriate resource. Following are some of the resources that you can use to help upskill your teams on Amazon OpenSearch Service:

- Immersion days – Our Solution Architects can deliver Immersion days, Hands-on workshops that are tailored to address the customer use cases, common implementation patterns, roadmap items that may be specifically related to use cases. that are run by AWS solutions architects.
- Hands-on workshops – self-service workshops built by AWS experts that can be followed by customers. Refer to https://aesworkshops.com/

- Whitepapers & Guides – AWS whitepapers are a great way to expand your knowledge of the cloud. Authored by Amazon Web Services (AWS) and the AWS community, they provide in-depth content that often addresses specific customer situations. Refer to https://aws.amazon.com/whitepapers/
- Blog posts – Published on AWS blog, written AWS experts and customers. These blog discuss latest announcements, best practices, solutions, service features, customer use cases and many other topics.
- Best practices – Online or conference talks, or sessions run by AWS experts that help customer understand best practices for Amazon OpenSearch Service.
- Our AWS Professional Services team can provide best practices and prescriptive advice, and a training program to help IT professionals understand and accomplish successful migrations.

Data Flow

Data Ingestion

Data ingestion is another key consideration during the migration. It focuses on how to get data into your Amazon OpenSearch Service domain. A thorough understanding of the data sources and formats is paramount when choosing the right ingestion framework for OpenSearch.

There are many different ways in which you can create or modernize your ingestion design. There is a rich ecosystem of open source tools that can be used to build a self-managed ingestion pipeline. OpenSearch Service supports integration **Fluentd** or **Logstash, OpenSearch Data prepper aggregators**. These tools are popular with most log analytics solutions developers. You can deploy these tools on an Amazon EC2 instance, or Amazon Elastic Kubernetes Service, or on-premises, whichever you find easier. Both Logstash and Fluentd support OpenSearch Service domains as an output destination. However, note that this will require maintaining, patching, testing, and keeping the Fluentd or Logstash software versions up to date.

You should be able to reduce your operational overhead by leveraging one of the AWS managed services that support integration with OpenSearch Service. For example, **Amazon Kinesis Data Firehose** is a fully managed service that helps build a serverless ingestion pipeline. Kinesis Data Firehose provides a secure and easy way to ingest, transform, and deliver streaming data to OpenSearch Service domains. It can automatically scale to match the throughput of your data, and requires no ongoing administration. It can also transform incoming records using AWS Lambda, compress, and batch the data before loading it into your OpenSearch Service domain. With a managed service you can retire your existing data ingestion pipeline, or you can augment your current setup to lower operational overhead.

Migration planning is a good time to assess whether your current ingestion pipeline meet the needs of current and future use cases. Generally if you are migrating from self managed Elasticsearch or OpenSearch cluster, your ingest pipeline should allow for easy swapping of endpoints from current cluster to Amazon OpenSearch Service domain endpoint with no or minimal client library updates.

Data Retention

When planning for data ingestion and storage. You should also plan and agree on **data retention**. For log analytics use cases, it is critical that you have right policies created within your domain to retire the historic data. When you are moving from an existing on-premises and cloud VM based architecture you could be using a particular type of instance for all your data nodes. Data nodes have same CPU, memory and storage profile. Most customers would configure high throughput storage to cater for their high-speed indexing requirement. This singular storage profile architecture is called *hot node* only architecture or simply hot-only. Hot-only architecture couples storage with the compute which implies

that you will need to add compute nodes if your storage requirement increases. Technically, you may not be needing additional compute if your data analysis or search requirement stays constant, however, the coupling of compute and storage forces you to do so. To decouple storage from compute, Amazon OpenSearch Service offers UltraWarm storage tier. UltraWarm provides a cost-effective way to store large volume of read-only data on Amazon OpenSearch Service by providing nodes that can address larger volume of data than traditional data nodes. UltraWarm is discussed in detail in Operational excellence section. During planning you decide the data retention and processing requirement. You should take advantage of UltraWarm tier to reduce the cost of your existing solution. Identify the retention requirement for your data and create ***Index state management policies*** to move data from hot to warm or to delete the data automatically from the domain when not needed. This also ensures that your domain does not run out of storage.

Data Migration Approaches

During the planning stage it is critical that you decide a particular data migration approach. Data migration approach dictates how you move your existing data that is in your current data store (elasticsearch/opensearch or otherwise) to the target store such that you do not have any gaps. We discuss in depth these strategies in data migration stage which is when you execute your approach. However, we will introduce these approaches here so you can make decision on appropriate data migration approach suitable to your journey during planning stage.

- **Build from snapshot –** Snapshots provide a way to backup your opensearch or elasticsearch data in a durable storage such as S3. Within this approach you take snapshot of your current elasticsearch or opensearch environment and restore the it in the target Amazon OpenSearch environment. After restoring the snapshot you can point your application to the new environment.
- **Build from source** – This approach implies that you are not going to move data from your current elasticsearch/opensearch cluster, instead you will reload the data directly from your log or product catalog source to the target Amazon OpenSearch Service domain. This is generally done with minor changes to existing data ingestion pipelines. In log analytics case, it may also mean you will reload the historical logs from your sources to new OpenSearch environment. For search use cases, it may require that you reload your full product catalog/content to the new Amazon OpenSearch Service domain.
- **Remote reindex from existing elasticsearch or opensearch environment –** this approach uses remote reindex API from Amazon OpenSearch Service. Remote reindex allows you to copy data directly from your existing on-premise/cloud based elasticsearch or opensearch domain to Amazon OpenSearch Service domain. You can build automation that can keep the data synchronize between two environment until you cut-over to the target environment.
- **Using opensource or third party data migration tools –** There are multiple open source tools available to migrate data from your existing elasticsearch environment to your target Amazon OpenSearch environment. One such example is logstash utility which allows you to extract data from an elasticsearch or opensearch cluster and copy it in Amazon OpenSearch Service domain. You may also find other third party tools that perform such a data migration.

We recommend that you evaluate all your options and opt for the one that you are most comfortable with. Remember to test all your tools and automation during your POC stage to ensure your selected approach is full-proof. Please refer to the data migration approach section under data migration stage in this document for details and step by step guidance on how to execute these approaches.

Deployment Frameworks

Many modern teams use continuous integration and continuous delivery practices and pipelines to automate the deployment of their solutions and infrastructure. If your team already uses CI/CD

pipelines, you should be able to easily incorporate Amazon OpenSearch Service in your ecosystem. If you are doing manual deployments now, should consider automating deployment and build pipelines to automate repeatable work, reduce operational overhead, and eliminate human errors.

Amazon OpenSearch Service can be deployed using a variety of open source infrastructure as a code frameworks such as **Terraform, Chef, and Puppet.** Terraform offers an OpenSearch module which can be used to create Amazon OpenSearch Service domains. In many cases you can use your existing infrastructure deployment pipeline and just replace the search engine module to point to the Amazon OpenSearch Service module.

If you are thinking about building pipelines from ground up or you prefer using AWS native services then there is a rich ecosystem of CI/CD tooling and services available. AWS offers **AWS CodePipeline, AWS CodeBuild, AWS CDK, AWS CloudFormation, AWS CodeDeploy and AWS CodeStar** services to automate infrastructure build, test and deployment. Deploying your pipelines using any of the above cloud native services has many advantages such as fully automated end-to-end (build, test, deployment) product releases, simple deployment to multiple environments (dev, test, pre-prod, prod), and seamless integration with other AWS services. Using these services, you can modernizing your deployment pipelines to automate deployments of OpenSearch Service domains across multiple environments.

Stage 2 – Proof of Concept

When doing a migration it is very critical to prove whether the target state solution will work as required. We strongly suggest running proof of concept exercise. This section focuses on the various aspects that need to be factored while executing a proof of concept.

Define entry and exit criteria

Having a clear entry and exit criteria is key to a successful POC execution. Define your entry criteria such as use case definition, access to environments, familiarity with various services or associated training requirements. Similarly define an exit criteria including functionality, performance requirements, security implementations etc. based on which you can evaluate the POC outcome.

Secure Funding

Based on the POC criteria definition, secure funding for the POC. Ensure you have performed the right sizing, and considered all costs associated. If you are migrating from on-premises to AWS, you should consider cost associated with migrating your frameworks over to AWS cloud from on-premises. If you're an existing AWS customer, work with your AWS account manager to understand if you qualify for credits that can be used for this effort.

Automate

Identify where all automation can be done and plan for a dedicated track to automate and time box the testing. Automate deployment and testing ensures you can rinse, repeat, test and validate without human errors at a rapid pace. By time boxing a tests you can ensure you are delivering on time and ready to pivot to other activities in case of challenges for e.g. if you plan to run performance tests and its taking longer than the estimated time then you can pause that activity and move to other tests and validation activities while your developers can fix the issues. You can come back to performance tests once the issues are resolved. Benchmark your existing solution performance and create automated performance tests that can be run to validate the effect of your configuration changes during the POC.

Thorough Testing

Exercise all portions of stack by making sure that you perform the required validations for all different layers such as ingest pipelines and query mechanisms that integrate with your Amazon OpenSearch Service domain. This will help you validate the end-to-end solution implementation.

Presentation Layer

In presentation layer, be sure to execute POC which spans across the below mentioned components.

- **Authenticate** - Validate the mechanisms that you are planning for authenticating your users.
- **Authorize** - Identify the authorization mechanisms that you want to follow and validate if those are working as expected
- **Query** - What are the most common use cases that you will encounter in production? What are some edge case scenarios that are critical to your business? identify these patterns and have them validated during POC.
- **Render** - Is the data being rendered accurately and appropriately for various users across platforms? For log analytics use cases you may want to build and test the dashboards in OpenSearch dashboard or Kibana depending up on the target version to see if it meets your requirements.

Ingestion Layer

In ingestion layer, be sure to evaluate various components such as Collet, Buffer, Aggregate and Storage.

- **Collect** - In case of Log Analytics use cases, validate whether all the data that we are logging is being collected. For search use cases, identify the sources that feeds the data and perform validations on completeness and correctness of data to make sure that collection phase has been executed successfully.
- **Buffer** – If you have a spiky traffic, you may want to make sure that you are buffering the data that is getting ingested. There are various ways which you can create buffering design e.g. collect data in Amazon Kinesis Firehose service or use S3 storage as buffer.
- **Aggregate** - Validate any aggregation of data that you perform during ingestion such as bulk API usage
- **Storage** - Validate whether the storage is able to optimally handle the ingestion that you are performing

POC Stages

We recommend you to follow the below mentioned stages in your POC to execute and validate the outcome. Don't be afraid to adjust the plan during the POC and iterate through these phases even though a lot of planning has been done upfront.

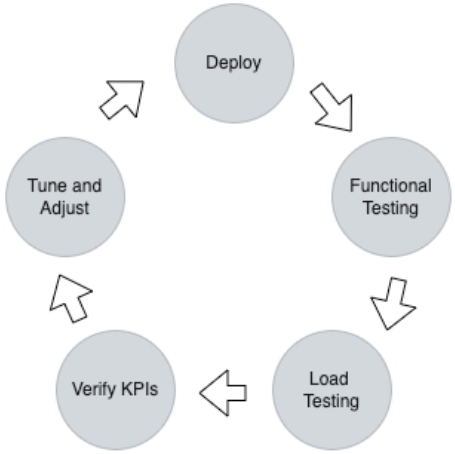


Figure 3 Proof of concept can be cyclic process

- **Functional Testing and Load testing –** Ensure that there is a thorough set of testing of all levels are being performed. Simulate failures in all potions of the stack. For e.g., if you have two large node cluster and one of them goes down, the other node has to take up all the traffic on your cluster. In such a scenario, having smaller but higher number of nodes will be a better choice so that the recovery from a node failure impact is more seamless. Test your workloads for peak loads and above to make sure that the performance is not impacted in such scenarios. While you are doing the testing, raise issues early and often so that any potential issues are being evaluated by various stakeholders at the right time.
- **Verify KPIs and Tune –** During the POC, ensure that you are meeting KPIs and meeting all the business outcomes that you have defined in your exit criteria of the POC. Tune the configurations in such a way that they are meeting the KPIs

- **Automate and Deploy –** Automation and monitoring are the other key aspects that need to be focused during the POC execution. Refine your automation steps and validate them along with detailed monitoring which gives all the stakeholders enough information to confidently evaluate the outcomes of POC. Have all the steps documented and create a runbook which can be reused for production roll-out.

Failure Simulation

We highly recommend you simulate failure scenario and validate whether your design offers resilience and fault tolerance required to meet your user requirements. You may want to simulate a failure of a data node to see if your cluster has enough resources to handle the recovery gracefully. You may want to simulate sudden burst of logs from some of your sources to see if you have right buffering settings in place to ensure your domain is not overwhelmed with large volume ingest. You should validate that your design is not exceeding any limits when you scale to a production deployment. You can refer to Amazon OpenSearch documentation to get more information on service limits.

Stage 3 – Deploy

By the time you reach this stage you must have completed your POC and have good idea about how you will go about deploying your target environment to production.

Validate Automation: During the deployment you exercise the automation you have created during the POC and validate that it's working as expected. You should also validate your CI/CD automation is working as expected when you make configuration code changes.

Verify Security: It is critical to verify all your security configurations are working as expected and your data is secure. Ensure solution is vetted against your company's security standards such as Identity provider integration, and whether your key users are able to login and access data that they are authorized to access.

Monitoring: Make sure you have tested your monitoring configurations and setup the recommended alerts. You should monitoring key metrics such as CPU, Memory, Disks, JVM, shard allocations etc. You can build dashboard in Amazon CloudWatch service to give you the insights your need about the health of your Amazon OpenSearch domain and associated integrations. You can verify that your operations support team have right access to these dashboard. The operational excellence section below provides a number of useful tips to ensure you are operating a high performant, resilient OpenSearch domain.

Exercise Alarms: Make sure you exercise all your alarms. If you are using Amazon CloudWatch or alerting plugin, validate that all integration such as SNS or chime or Slack integration is working as expected. You should simulate alerts to ensure the alerts are delivering correctly to the destination channel, and alert text have right information e.g. a link to associated runbook for your support team to execute an associated remediation process.

Stage 4 - Data Migration

Now that your target environment is ready you can execute your data migration strategy. It is expected that you have assessed various migration approaches during the planning stage.

Migration Approaches

In this section we are going to talk about different ways and patterns that you could use to migrate an Elasticsearch cluster to Amazon OpenSearch Service. Following are the list of factors (not-exhaustive) to consider for choosing a pattern

- Whether you want to copy data from existing self-managed cluster or you are rebuilding from original data source (log files, product catalog database).
- Version compatibility of the source Elasticsearch or OpenSearch cluster and target Amazon OpenSearch Service domain
- Applications and services dependent on the Elasticsearch or OpenSearch cluster and available window for the migration.
- Volume of the indexed data in your existing environment.

At high level there are 4 different patterns, each discussed in detail below.

1. Building from snapshot

Snapshots are the most popular way to migrate from a self-managed Elasticsearch cluster to Amazon OpenSearch Service. This approach allows you to copy data from source Elasticsearch/OpenSearch cluster to target Amazon OpenSearch Service domain following few simple steps. This approach is simple and useful if your source and target are compatible, you have a large volume of indexed data in existing cluster which can be time consuming to reindex, or your source data is not available at all for reindexing. Broadly, the snapshot-restore process consists of the following steps:

1. Take a snapshot of the necessary data (indices) from the existing cluster, and upload the snapshot to an Amazon S3 bucket.
2. Create an Amazon OpenSearch Service domain.
3. Give Amazon OpenSearch Service permissions to access the bucket, and give your user account permissions to work with snapshots. Create a snapshot repository and point that to your bucket.
4. Restore the snapshot on the Amazon OpenSearch Service domain.
5. Point your client applications to new Amazon OpenSearch Service domain.
6. Create ISM policies for configuring retention (optional)

Snapshots are incremental, therefore a snapshot can be run and restored incrementally. Snapshot allows you to extract data in bulk in the form of files on a storage system (e.g. S3), and the load these files in target environment through restore API. This eliminates the need for reindexing which is time consuming, and it also reduces the network traffic.

Considerations

Following are things to consider when using snapshot-restore approach.

1. You cannot search or index data into an index while it is being restored. However, you can search and index an index that is being taken snapshot of.
2. The source and target Elasticsearch/OpenSearch versions must be compatible. A snapshot of an index that was created in:
    - 5.x can be restored to 6.x
    - 2.x can be restored to 5.x
    - 1.x can be restored to 2.x
3. Because this is a point-in-time restoration of the Elasticsearch snapshot, subsequent changes in the source cluster won't be replicated to the target Amazon OpenSearch Service domain. You can stop ingestion of the data into the source Elasticsearch cluster until the restoration is done or you can repeat the snapshot restore process a few times, since it is incremental, only the changes will be copied and restored in target environment in less time than first restore. Once restoration is successfully finished, you may repoint the ingestion application(s) to the newly built Amazon OpenSearch Service domain.

4. Snapshotting by default includes snapshot of the cluster state, and all indices. When migrating from Elasticsearch, you may have to create equivalent Index lifecycle policies in the target environment using Index state management feature in OpenSearch. ILM is not supported in Amazon OpenSearch Service.
5. You cannot restore a snapshot to an earlier version of Elasticsearch or OpenSearch e.g. you cannot restore snapshot of version 7.10 to 7.9
6. You export snapshot to a designated storage location called repository. Elasticsearch/OpenSearch creates a number of files in the repository. You cannot modify or delete these files. Doing so may fail the restore process or create inconsistencies.

2. Building from source

As described earlier, build from source is the approach where you do not migrate data from current Elasticsearch or OpenSearch environment instead you build indices in target domain directly from your log, or product catalog data source or content source. There can be various reasons to do so e.g.

- Your source and target environment versions are not compatible for snapshot restore.
- You want to change your data model in the target environment as part of the migration.
- You want to jump to most recent version of Amazon OpenSearch Service to avoid rolling upgrades, and you want to address the breaking changes in one go. This can be a good idea if you are self-managing a relatively older version (5.x or lower) of Elasticsearch.
- You may want to change your indexing strategy e.g. instead of rolling over every day, you may rollover every month in the new environment.

Once you have decided to build from source. Following are different ways to achieve it, depending upon the type of source of your data.

1) Using Amazon Database Migration Service

If the source of your data is an RDBMS and if it is a supported source in Amazon Database Migration Service (DMS) then you can use DMS to copy data from your data source to your target Amazon OpenSearch Service domain. DMS supports full load and Change Data Capture (CDC) options. In full load option the DMS task copies all data from source database table to a target OpenSearch index. You can use default mapping or provide custom mapping configurations. In CDC mode, DMS first makes full copy of the source table records into target index and after then captures changed data (updates/inserts) and copies it to the OpenSearch index. For more information you can refer to the following blogs. https://aws.amazon.com/blogs/database/introducing-amazon-elasticsearch-service-as-a-target-in-aws-database-migration-service/ and https://aws.amazon.com/blogs/database/scale-amazon-elasticsearch-service-for-aws-database-migration-service-migrations/

2) Build from document source

If your data source is not an RDBMS or it is not supported by DMS then you may have to create a custom solution using open source tools or a combination of open source tools and AWS native services. Your source data has to be converted to JSON documents before it can be loaded in OpenSearch. Note that in most cases customers already have pipelines setup from their source to current Elasticsearch or OpenSearch environment, and it is just a matter pointing those data pipelines to OpenSearch with appropriate changes in client libraries and (if required) data model changes in indices in Amazon OpenSearch Service domain. Following section discusses few key things to keep in mind while building indices from source.

- ***Location of the documents*** **-** The documents could already be available in AWS cloud, in an Object storage such as Amazon S3 or it might be stored in an on-premises storage location such as file system.
- ***Format of the documents*** **-** The documents could be already in JSON format ready to be ingested into Amazon OpenSearch Service domain or it might need to be cleansed, processed and formatted (into JSON) before it can be ingested into Amazon OpenSearch Service domain.

Following are high level steps

- Define index mapping and settings in Amazon OpenSearch Service domain
- Extract data from the document source and copy it into an object storage e.g. S3. This could be done with an open source tool (e.g. logstash), AWS service client (Kinesis agent) a third party commercial tool, or a custom program.
- Configure an opensource tool (e.g. logstash, fluentbit) or a native AWS service (e.g. AWS Lambda, Amazon DMS) to convert data into JSON documents and load it periodically or continuously from object store to opensearch.

Following doc link provide additional information

https://docs.aws.amazon.com/opensearch-service/latest/developerguide/integrations.html

3. Remote reindexing

In this case the Indices of the source self-managed Elasticsearch is migrated into Amazon OpenSearch Service domain using *reindex API*. Reindex API allows you to create an index from existing OpenSearch/Elasticsearch index. The existing Elasticsearch/OpenSearch index could be in the same cluster where reindex operation is executed, or it could be in a remote cluster. Amazon OpenSearch Service support reindex API with remote clusters. You can reindex from an index in a self-managed Elasticsearch to an index in Amazon OpenSearch Service. Remote reindex supports Elasticsearch 1.5 and higher for the remote Elasticsearch cluster and Amazon OpenSearch Service 6.7 and higher for the local domain. Please follow the blog post to learn how you can perform the same - https://aws.amazon.com/blogs/big-data/migrate-data-into-amazon-es-using-remote-reindex/

4. Using Logstash

Logstash is an opensource data processing tool that can collect data from source, perform transformation/filtering and send data to one or more destinations. Logstash offers *logstash-input-elasticsearch* and *logstash-input-opensearch* input and logstash-output-opensearch output plugins which can read data from Elasticsearch/OpenSearch and write data to Amazon OpenSearch Service domain respectively. More information on how to use Logstash for loading data into Amazon OpenSearch Service can be found at this link from Amazon OpenSearch Service docs. To get more detail about logstash-input-opensearch plugin please read blog here.

Stage 5 – Cut Over

This stage discusses various methods you can employ to cut over from your current Elasticsearch or OpenSearch environment to the target Amazon OpenSearch Service domain. Cut over can be done in two steps. We establish a data synchronization mechanism to keep target environment synchronized with source and then we perform the swap from current to the target environment with or without downtime.

Data Synchronization

For any system receiving continuous data, data migration may require that you stop receiving new data during the migration and execute migration in maintenance window (with possible downtime). If you cannot afford downtime then you capture changes after you have triggered the migration. You replay the changes on the target to keep it updated and synchronized with source until you perform the cutover. Following section discusses various ways we have seen customers keep source and target synchronized.

Log Analytics workloads

For log analytics workloads you may do update sync in various ways

- You may run two environments side by side until the retention period and run ingestion to both current and target environment. At some point in time, you may decide to cut over and point your applications to the new environment. Sometimes you may ingest new data from the log or document sources to both existing cluster and target OpenSearch environments, and back fill the older data in the target environment by copying it from the current environment. In all cases you have to make sure you do not leave any gap in your data that impacts your users.
- Prior to the data migration, you could decide to pause your ingestion to the existing environment. However, this approach means your users may not be able to search latest or changed data from your existing environment until your data migration is completed. Once you data migration is completed, you can point your data ingestion to target environment and switch your applications and clients over to the target environment. This means no new data will be available until migration is completed, however, the system will remain available for search. You should have means to hold source logs/data in your sources until new environment is available.
- You may continue to use the current log search engine until your first pass of data is migrated. Then you backfill the remaining data that has been produced since the first pass was initiated. Assuming the remaining data is much smaller than the first pass, therefore you may be able to pause ingestion while your remaining data is synchronized which may be a few minutes or few hours period. You can also perform few passes of this approach until your synchronization window becomes small enough to pause ingestion from source to current environment and cutover to the target environment without impacting your users. This approach is depicted in the diagram below.

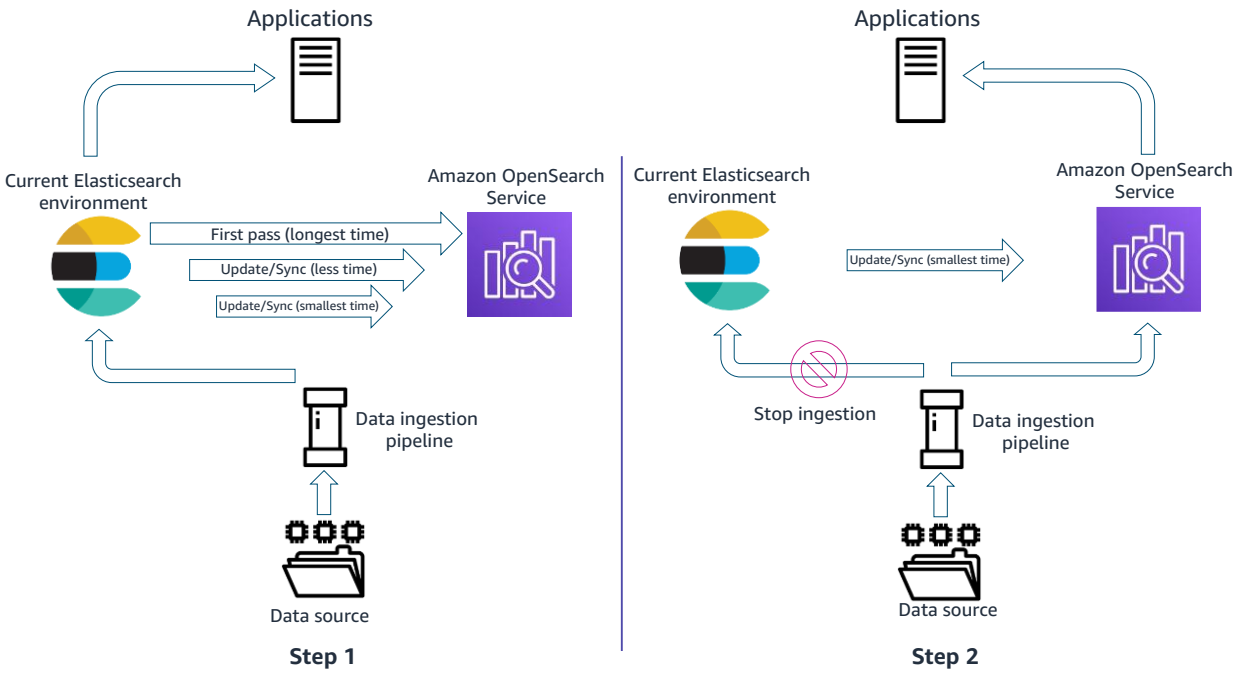


Figure 4 Using incremental snapshot and restore to update/sync data

In all the ways given above, you have to ensure your target has all data up to date before you perform the cut over. Following are couple of workload specific update/sync suggestions

Search workloads

- For search workloads typically you will pause the ingestion from source to the current environment. You will copy all your data from current environment to the target environment and you will put in place a change data capture mechanism which can determine what data has changed since the start of migration. You will then copy the changed data to the Amazon OpenSearch environment. In most cases the search applications data ingestion pipelines already have CDC mechanism built in, and it usually is a matter of pointing your pipeline to the new environment after the data is migrated from the current environment.

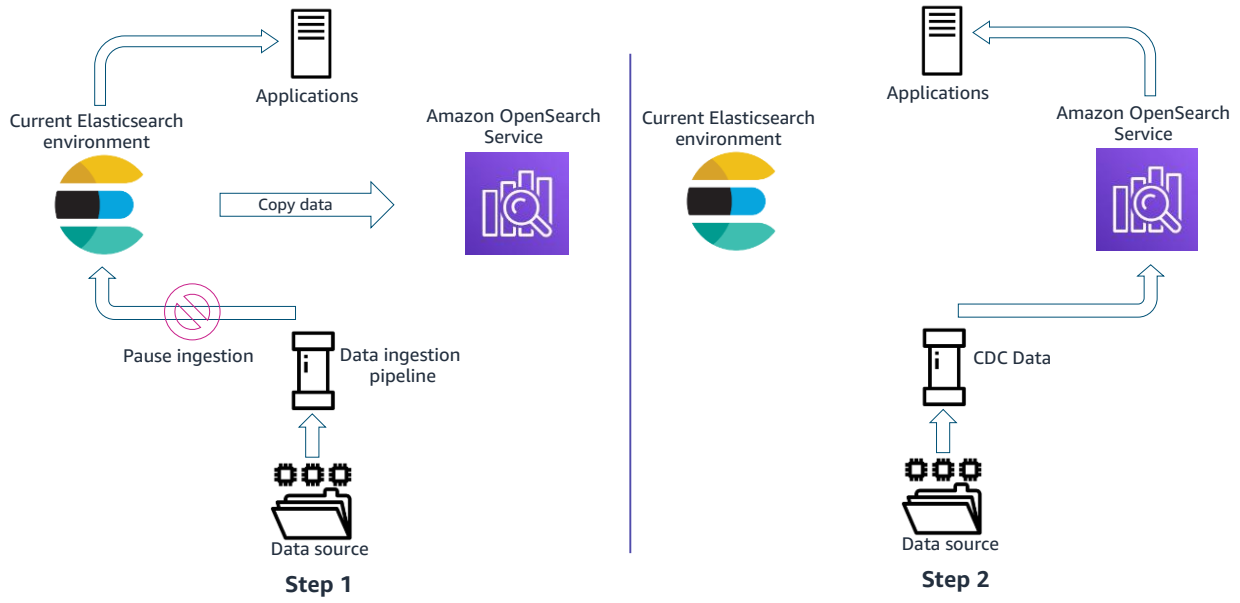


Figure 5 building index entirely from source for search use cases

- Some search workloads only require loading full data from source database/data source to new OpenSearch environment. Once the load is completed, the client applications can cut over the new environment. This is the simplest way to achieve migration for search workloads.

Swap/Cut-over

Final step in the migration journey is the swapping to the new environment. It is one of the critical phases. At this point, you are ready to go live. You have data synchronized up to date, you have monitoring and alerts configured, your runbooks are up to date and you are ready to switch over. You have to make sure your ingestion is flowing normally, your metrics are healthy from your new environment. During this stage you will plan and execute the switching over of the client connections from your existing elasticsearch/opensearch environment to the new Amazon OpenSearch Service cluster. Be mindful of the client library changes that may be required, at this point you should have tested all your client functionality in your lower environments with the Amazon OpenSearch Service for compatibility and performance.

You may have a client application that may need to point to the new environment. For that you may update the DNS entry from old environment to the new environment and closely monitor your application behavior to ensure your users are getting the right experience.

We recommend that you keep your older environment up to date so that it can act as a fall-back in case you encountered any problems with the new environment. Generally, if you have followed the guidelines in this document, you will have a safe switch over, however, for business continuity purposes you will need to keep your existing environment around for some time.

We have seen customers continue to operate the both the environment for few weeks after the swap before decommissioning the older environment. We recommend you choose an appropriate strategy that lines up with your business continuity requirements.

Stage 6 – Operational Best Practices

Amazon OpenSearch Service documentation has a dedicated section on Operational best practices. We recommend that you follow guidance provided in the docs to operate your newly migrated environment.

Summary

Amazon OpenSearch Service takes away the undifferentiated heavy lifting that is required to develop and operate self-managed Elasticsearch or OpenSearch clusters. Customers thinking to migrate to Amazon OpenSearch Service can use the process outlined in this guide to plan and decide a migration strategy that works best for their situation. Migrations can be as simple as taking a snapshot from self-managed cluster, and restoring it in Amazon OpenSearch Service domain, or it could be an involved process requiring you to test all existing functionality and integrations. This guide provides information that can be used by migration project teams to make sure they have covered all aspects of a migration and build a robust strategy to execute.

Contributors

Contributors to this document include:

- Muhammad Ali
- Soujanya Konka
- Gene Alpert
- Sung-il Kim
- Hajer Bouafif
- Kevin Fallis
- Jon Handler
- Prashant Agrawal
- Muthu Pitchaimani
- Kunal Kusoorkar
- Imtiaz Sayed
- Ina Felsheim
- Marc Clark
- Bob Taylor
- Aneesh Chandra PN

Document Revisions

Date	Description
Jan 2023	First publication

References

- Elasticsearch Snapshot and Restore reference
- S3 Repository Plugin
- Working with Amazon OpenSearch Service Index Snapshots
- Use Amazon S3 to Store a Single Amazon OpenSearch Service Index (blog post)
- Elasticsearch Repository Settings: Recommended S3 Permissions
- Elasticsearch Client Settings