elastic

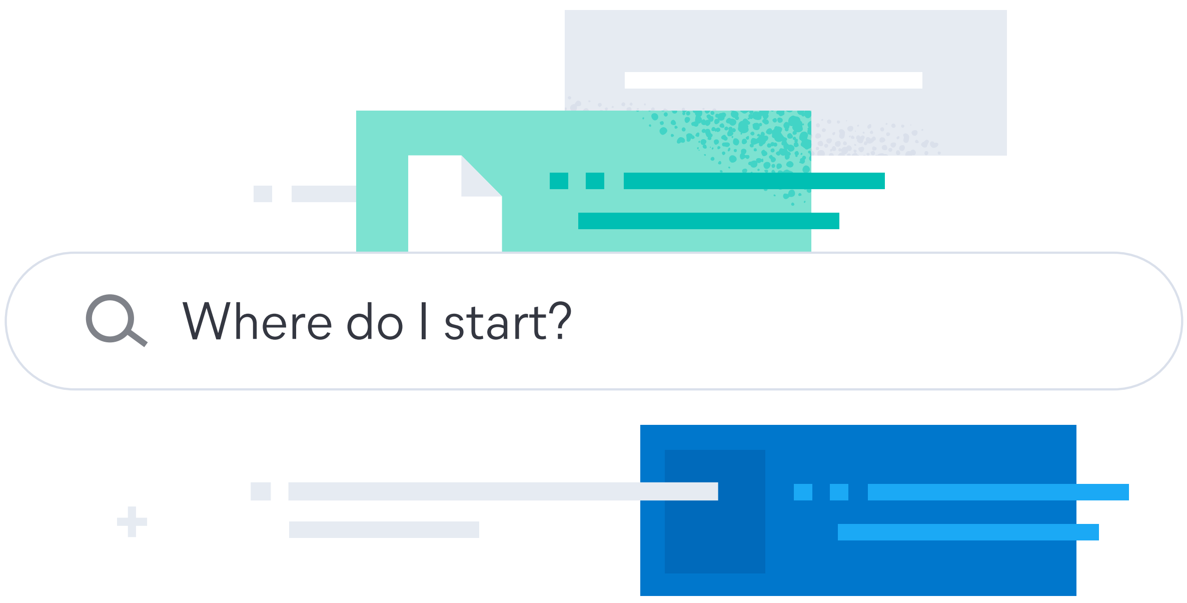
Search. Observe. Protect.

Migrating to the Elastic Stack

Strategy for speed, scale, relevance

elastic.co

Table of Contents



Introduction

A software platform change can seem like a daunting task. Migrating to new systems requires defining the right design, procuring the cloud resources or hardware for self-managed software, security review and implementation, deployment and testing, governance policies, data and other content transfer, stakeholder support, staffing, training, and more. And even when the objectives of enhanced data insight, more scalable technology, and lower total cost of ownership (TCO) are recognized, the very process of migrating to a new platform can interrupt even the best of plans.

This guide is intended to simplify your migration to the Elastic Stack. We'll walk through strategies for moving away from legacy platforms such as Splunk® and ArcSight® while minimizing disruption, risk, and cost.

Why Elastic?

Elastic is a search company, meaning we make the power of search — the ability to instantly find relevant information and insights from large amounts of data — available for a diverse set of applications.

The Elastic Stack is a powerful set of products that ingest and store data from any source, in any format, and perform search, analysis, and visualization in milliseconds or less. We have also built solutions on the Elastic Stack that address a wide variety of use cases: Elastic Enterprise Search for workplace search, app search, and site search; Elastic Observability for logging, metrics, and application performance monitoring (APM); and Elastic Security for security information and event management (SIEM) and endpoint security.

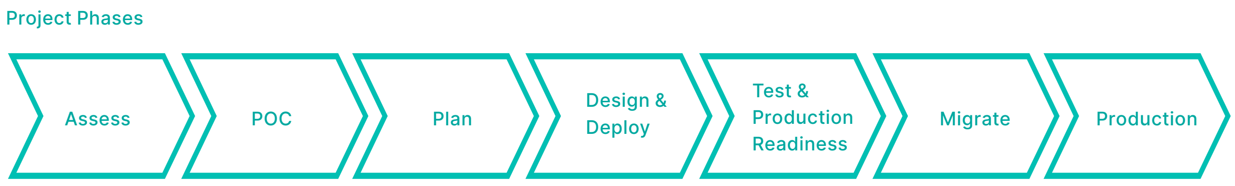
The Elastic Stack and our solutions are designed to run in public or private clouds, in hybrid environments, or in traditional on-premises environments. As the technology landscape shifts, our products grow and adapt. In that sense, we believe that our company is truly elastic.

Planning a migration

In any migration project, it's important to demonstrate business success early on. When migrating to the Elastic Stack, this means focusing on the highest-priority data sources and insights first and building out visualizations and dashboards to meet business objectives. Over time, additional data sources can be added. The distributed nature of the Elastic Stack makes it easy to scale up and add data nodes at any time.

We recommend breaking your migration journey into the following phases. The Project Owner or Business Stakeholder should be aware that while each of these phases may differ in duration and detail in your organization, it's important to ensure that each milestone is completed to the level that's right for your organization:

- Assessment
- Proof of concept
- Planning
- Design and deployment
- Testing and production readiness
- Migration
- Production



Assessment

The assessment phase of the project should include identifying and prioritizing the data sources, visualizations, and team and organizational needs. This phase can be broken down into:

- **Business assessment:** Meet with each of the business teams to determine priority and end user needs, including outcomes, functionality, and training. Also includes requirements such as regulatory compliance, policies, data retention requirements, and disaster recovery RTO/RPO.
- **Technical assessment:** Meet with technical stakeholders in various teams to discover their existing environment, architecture, and data sources (source type and format, complexity). Use this information to begin designing the new environment and rule out any potential roadblocks.
- **Deliverables:** A document listing the success factors and business objectives and a description of the current environment, followed by a detailed list of business and technical requirements.

Proof of concept (POC)

We recommend including a POC phase in your migration journey if it wasn't already performed as a part of the evaluation process. A POC environment with a small number of data sources that represent high-priority business needs can be beneficial for initial development and validation and can streamline future phases. The POC environment can also be useful for presenting and demonstrating value to business owners and sponsors and can be used for testing the addition of new data sources further down the road.

Planning

The planning phase should include:

- Defining the implementation tasks and milestones
- Estimating the effort for each task and milestone
- Describing the dependencies (business, technical, people)
- Identifying the main stakeholders' roles in the project and allocating tasks
- Creating and sharing a project plan

Throughout the duration of the project, the Project Management (PM) team should validate the planning requirements against the progress of the project. Common PM tasks and checkpoints may include the following:

- Continuous consulting milestone reviews
- Dependency callouts
- Risk escalations
- Change request management
- Project status report to stakeholders
- Resources and budget management
- Handover, user training, and data onboarding

Design and deployment

Design best practices

The first step is to architect an Elastic Stack platform that is limited in initial size while being designed for scalable growth. There is no need to overarchitect a distributed platform initially — the Elastic Stack or Elastic Cloud can be set up for future growth using a handful of simple design principles:

- Use a best-practice architecture from the beginning where dedicated master-eligible nodes allow for the future addition of data nodes as needed for growth
- Implement local storage per node that is not dependent on a centralized storage system
- Establish guidelines up front for relative performance requirements of current data versus historical data in a hot-warm-frozen index strategy — where current data resides on high-performing SSD storage while older data is stored on denser spinning disk — in order to minimize TCO and maximize data density
- Use the Elastic Common Schema (ECS) with Elastic Security and Beats to standardize the data mapping and simplify the ingestion process via Filebeat templates
- Use a data stream that centralizes all data first into high-volume Logstash pipelines or other tools such as Kafka or Redis, and then allow the various destination platforms to tap into some or all of that data to enable flexibility and future growth; bifurcation of the data stream can then be used to incrementally migrate workloads to the Elastic Stack

- Use ingest nodes to enrich or transform data during ingestion; a pipeline with multiple processors can be configured per data source as needed to ensure data consistency from migration through to normal production data streaming
- Design for high availability and fault tolerance with multiple zone availability using Elastic Cloud availability zones, or multiple cluster distribution using cross-cluster replication and/or cross-cluster search
- Create a test/staging environment — even if just for a single node — where business teams can test ingestion and mapping; mapping settings can then be moved into their production instance via Elasticsearch index templates, allowing you to disable dynamic mapping in production while allowing it in the test or staging environment (dynamic mapping is a great tool for experimenting or developing with the Elastic Stack and makes it easier to get an initial understanding of your data, but it should not generally be used in production due to potential problems with field explosion or performance degradation)

Optimizing for your use case

One key decision that arises in all customer platforms is designing for your organization's specific requirements. Parameters such as cost and query response rate can be tuned to your particular needs. Elastic's hot-warm architecture and Elastic Cloud deployment templates are based on proven and repeatable patterns and make it easy to select the right architecture for your deployment.

As an example, perhaps you need to optimize for a write-heavy environment with terabytes per day of log and endpoint ingestion, but the number of searches on the platform are moderate. In this situation, you may choose to increase the amount of data stored per node to maximize data density, while recognizing that complex searches may require minutes rather than seconds. An opposite example might be a search environment for an ecommerce catalog — you may support millions of search queries per minute, but the data changes daily and thus is a read-heavy platform that requires more CPU compute resources and memory while utilizing only moderate input/output (I/O). Or perhaps you have a multi-use platform that is supporting multiple use cases and requires a balance of both read and write operations, with multiple clusters within the same center of excellence (CoE).

These requirements, identified early, can lead to good decisions that will allow the platform to scale into the future. High I/O requirements and expectations for near real-time search require investment in the hot tier of storage with <4TB per node. For more intermittent queries or when allowing longer response times, higher data density of 8-16TB per node is possible with warm or frozen nodes. Nodes can always be added for scale, and different node types allow the right blend of hot, warm, and frozen nodes to maintain an acceptable cost-to-performance balance. Rollups can be utilized to store key data results while reducing the volume of high-frequency data. Finally, index lifecycle management (ILM) can be used to define index lifecycle policies to roll over older indices to lower-cost storage or delete old data in a predefined way.

Designing with operational simplicity in mind

There are many options for simplifying the long-term administration and growth of your Elastic Stack platform. While this guide will not go into great depth on administration and operations, here are some deployment options that are commonly used to ease administration, improve team efficiency, reduce current TCO, and enable the environment for future growth.

- Elastic Cloud eliminates the need to run servers locally. Adding new nodes is as simple as increasing the total memory, storage, and compute resources and can be optimized for a hot-warm architecture and your solution requirements. Backend storage in Google Cloud, Microsoft Azure, or Amazon Web Services (AWS) can be selected at time of cluster creation.

- Elastic Cloud Enterprise (ECE) provides an on-premises or cloud-based orchestration product based on the same technology foundation used in Elastic Cloud. ECE provides Elasticsearch, Kibana, and an administrative UI for Docker-based containers.

- Elastic Cloud for Kubernetes (ECK) simplifies setup and administration with Elasticsearch, Kibana, and an administrative UI for Kubernetes. It is based on the Kubernetes Operator pattern and is optimized for managing one or more deployments on Kubernetes.

Other platform orchestration tools are also available. Orchestration, or the automation of the installation, configuration, and provisioning of nodes or clusters, can greatly improve the speed and consistency of administering a large Elastic Stack environment, as well as the efficiency of the CoE operations team. There are a few different legacy methods of orchestrating the Elastic Stack:

- Official Ansible playbook for Elasticsearch, provided by Elastic
- Official Puppet module for Elasticsearch, provided by Elastic
- Official Chef cookbook for Elasticsearch, provided by Elastic
- Docker images for Elastic

Deploying the Elastic Stack

Designing and optimizing your Elastic Stack deployment to fit your needs from the beginning will help you avoid redoing work in later stages. In this section, we'll share several simple practices and considerations that will help you set yourself up for scalable growth.

- Deploy a development environment — a pre-production platform that mirrors production and is ready for pipeline ingestion, building mapping templates, and use case development tasks. The development environment is a valuable staging platform for onboarding each new data source, as well as implementing and testing any new features, dashboards, plugins, or development. The vast collection of Elastic integrations, Elasticsearch plugins, Kibana plugins, and Elasticsearch Query DSL API libraries can simplify the data onboarding step.

- Deploy the production environment — a platform ready to support the production load that can be easily scaled over time by simply adding data and ingest nodes.
- Deploy the data streaming platform — a pipeline that is easily controlled and flexible enough to meet the needs of multiple teams, and with the bandwidth to ingest events, logs, or observability data through the PUSH vs. PULL model.
- Use machine learning jobs and automated anomaly detection across all solutions to automate your monitoring and anomaly detection, and increase visibility and reduce the time needed to identify and respond to anomalies and threats.

Automation

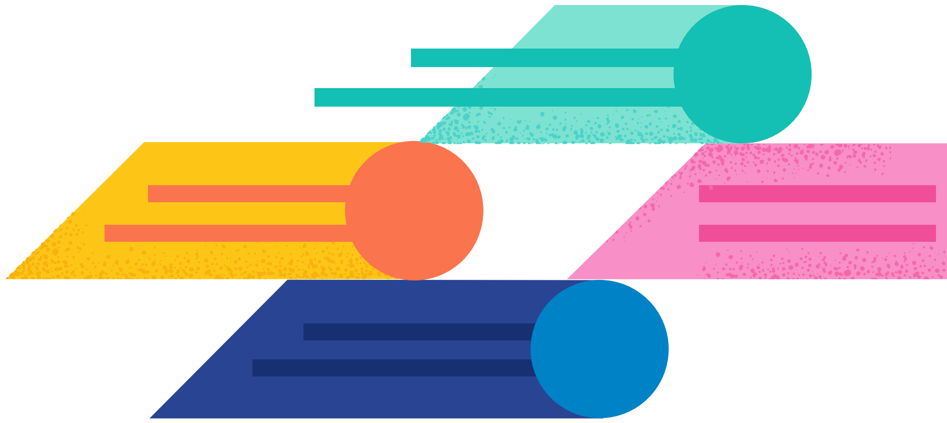
Elastic machine learning features can identify and alert on anomalies to reduce your dependence on dashboards, while alerting can be integrated with third-party systems such as email, PagerDuty, and Slack to enable proactive notifications. Elastic REST APIs and the Elasticsearch Query DSL allow automation of many areas of the Elastic Stack:

- Orchestration / build
- Upgrade
- Operational health / diagnostics
- Source control / use case deployment
- Machine learning
- Alerts
- Index lifecycle management
- Data enrichment via ingest processors or Logstash
- Index templates
- Use of ECE and ECK for rapid cluster deployment and zone management
- Multi-zone availability with ECE and ECK
- Endpoint detection and response (EDR)

Testing and production readiness

Testing should happen across the project at various levels.

- User testing for each high-priority use case: User testing often includes loading representative sample data into the platform. The expected outcome is that the templates, data mapping, dashboards, and alerts for high-priority data are ready at launch.
- Data pipeline benchmarking: By streaming the full data pipeline into the Elastic Stack, the ingest pipeline can be tested for throughput and resiliency. This data can be easily deleted prior to the migration cutover. The expected outcome is a data pipeline utilization report that can be used as a baseline for future growth (max throughput, max EPS, average CPU, memory, disk consumption based on customer's log sources).
- Elasticsearch performance benchmarking: While traditional benchmark sizing is not necessarily required in a highly distributed system, since nodes can be added as needed to scale with growth, a benchmark can be utilized to optimize data volume per node and reduce long-term TCO. Elastic benchmark results are available here, or you can capacity test in your own environment with Elastic's open-source benchmark framework, Rally.
- Testing of a disaster recovery plan that incorporates zone or site-level failure: The expected outcome is that the team has a well-documented plan and understands the steps required to recover from critical zone or data center loss.



Migration

With your Elastic deployment ready, it's now time to start migrating content over.

- Observability data: Using lightweight Beats and/or Logstash, log event sources and metrics in the ECS format can be ingested to quickly and predictably map data in a standardized and easily visualized format in the Elastic Stack. Elastic APM agents provide application-level monitoring and traces.
- Security data: Traditional logs from firewalls and network routers ingested by Elastic Security and Beats, plus packet capture data from Packetbeat, allow correlation of security events and threats. The Elastic Security detection engine provides out-of-the-box rules for common threats. This makes it possible to have all security-related data within a single window, with correlation across all host and network nodes.
- Data from legacy agents: You can use legacy shipping agents such as Splunk forwarders or ArcSight SmartConnectors to redirect data directly into the Elastic Stack. This is a quick method to stream data from your pipeline into Elasticsearch.

Migration approaches

In general, a migration from a legacy platform is most effective when migrating directly from the original raw data (logs, metrics, security events, etc.). Migrating processed data or summary rollups from legacy platforms is far more complex and less useful. Based on the data retention duration required, migrating the original raw data logs will determine the transition period when both environments are active in parallel, which reduces the impact to the legacy platform and reduces the overall risk and complexity of the migration. Various migration approaches based on raw data access and ingest are described below.

Bifurcation approach

With the bifurcation approach, the traffic from legacy agents is redirected to Logstash or Syslog mechanisms to intercept the events to Elasticsearch. Once the use cases under Elastic management have been approved, the legacy agents can be switched to Beats.

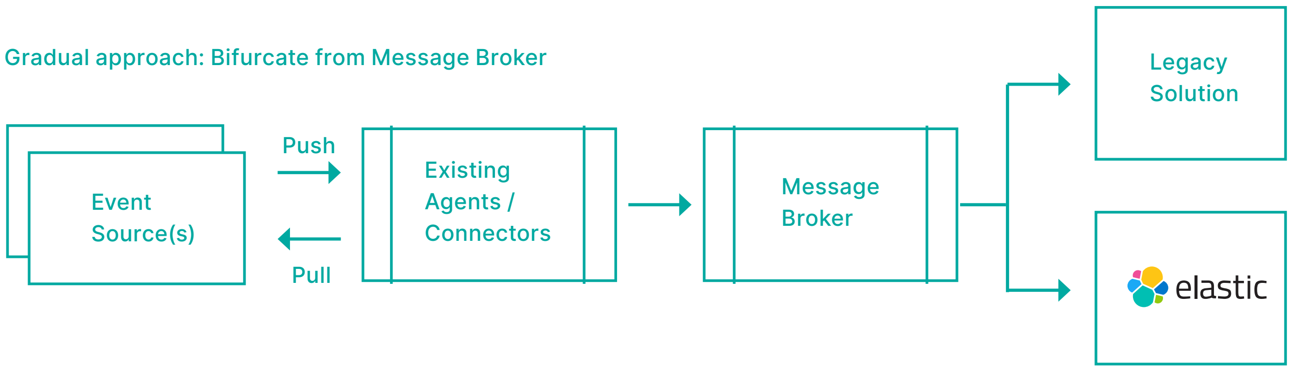
Advantages:

- Allows time to understand the implications of replacing legacy agents (such as Splunk universal or heavy forwarders)

Disadvantages:

- Requires configuration/changes to the destination from legacy agents
- Licensing on both services continues during the transition

Gradual approach: Bifurcate from Message Broker



Cutover approach

The cutover approach introduces the installation of all security events or log collectors with Elastic Security or Beats to forward events to Elasticsearch.

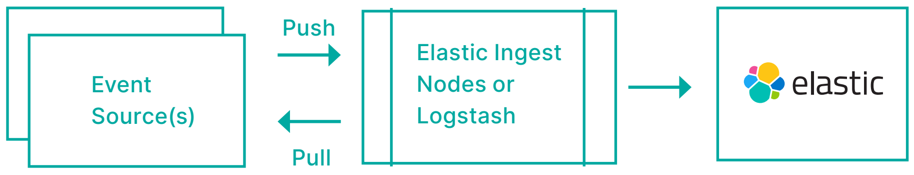
Advantages:

- Clean process with no dependencies on the legacy service
- Can use Beats for rapid deployment or existing tools such as Syslog with Logstash; these can sometimes be deployed in parallel with legacy agents

Disadvantages:

- Requires changes on all edge log collectors from the beginning

Direct approach: Cutover from Event Source(s)



Replication approach

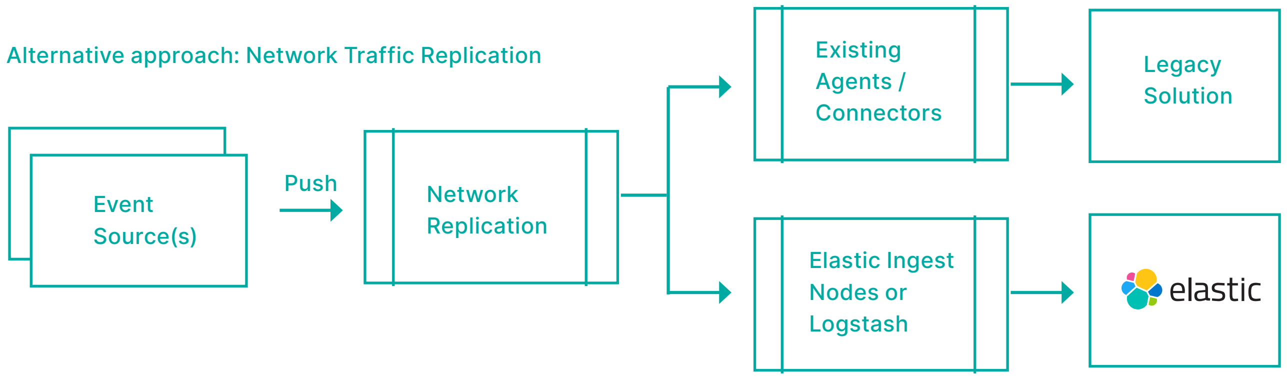
With the replication approach, the traffic from the legacy agents is redirected to a load balancer or message broker with traffic replication capability.

Advantages:

- No change to hosts or applications that already have legacy agents

Disadvantages:

- Adds operational overhead (licensing/hardware spend on traffic replication capabilities)
- Requires changes to the destination from legacy agents


Alternative approach: Network Traffic Replication
Event Source(s)
Push
Network Replication
Existing Agents / Connectors
Legacy Solution
Elastic Ingest Nodes or Logstash
elastic

Prioritizing data sources

An organization may have hundreds or thousands of data sources. Your organization knows best which data is most important, and it is critical to identify the highest-priority sources so that these can be implemented first.

Often, certain data sources are simply more important than others. By breaking down the process of migrating data sources based on priority or volume and standardizing the data mapping to ECS, a rolling migration process can be implemented that allows for one data source or a small subset of data sources to be moved per sprint. This follows a common pattern, as follows:

1. Map the data source first into a test or development environment, and verify it works
2. Copy the index template to the production environment, and ingest the data source into production
3. Build out visualizations and dashboards for the new data source and confirm that the data is mapped correctly and the resulting dashboards provide the required insights
4. Decommission the legacy system that provided data tools for that data source (and reuse the nodes or hardware if possible — these could potentially even be wiped, rebuilt, and added to the Elastic cluster for additional capacity)
5. Review and repeat for remaining data sources

Migrating searches, dashboards, and alerts

A large organization may maintain hundreds or thousands of visualizations, dashboards, alerts, and saved searches. Many of these are created by individual teams focused on their unique data requirements or area of monitoring. There are a few strategies commonly used for migrating these standard queries:

- As discussed earlier, prioritization is a key technique. Many existing visualizations or alerts were needed at one point in time, but may not be frequently used or current. Each data team should be asked to prioritize the most commonly referenced dashboards.
- Visualizations can be developed relatively quickly by the analyst teams. Kibana Lens provides a drag-and-drop visualization creator that provides intuitive and rapid implementation of new dashboards.
- Elastic Stack alerts can be created quickly with Alerting and Actions in Kibana. Less precise and sophisticated alerts can also be developed using the Elasticsearch Query DSL via Watcher. For many common use cases, Alerting and Actions provides quicker implementation.
- Validated dashboards, alerts, and reports built in a test environment can also be migrated using a REST API or by exporting to the production environment.
- Given the breadth of monitoring needs, enabling your teams to develop their own dashboards and alerts is critical. The Data Analysis with Kibana course covers the skills your teams need to create rich dashboards quickly.

Production

The production phase is a result of the success of the preceding steps. Long-term operation for stability and growth requires monitoring and adding new data nodes as needed, and managing the environment using good onboarding techniques such as planning for additional volume and static index templates.

Operational tips

Operational efficiency can be gained through simple techniques to ensure consistency for your users using simple automation and abstraction:

- Centralized Stack Monitoring of the Elastic environment to understand usage capacity and performance, including tracking the percentage of data integrated/migrated
- Index automated rollover using ILM
- Centralized pipeline management for input, processing, and output
- Wildcards (*) on index patterns for querying multiple indices
- Index aliases to abstract index names

Center of excellence

The key to building a production CoE is a well-trained and committed Operations team that focuses on the stability and predictability of the platform while allowing the data teams to consistently enhance their use of the platform and achieve their business needs. Low TCO is attained by growing the capacity of the platform as needed to maintain the required response times and adding data nodes over time. With the distributed nature of the Elastic Stack, there is no need to overbuild the platform initially — new data nodes can be added to the cluster in real time to add capacity resources. This is the core of Elastic's resource-based pricing, and can be optimized to your organization's needs based on the total volume of data stored, data retention, and response time requirements.

Building your Elastic Team

When building and administering the Elastic Stack environment, we commonly see a few key roles or personas within teams. Within a small team, it is certainly possible to have one person doing all of the following tasks; within a larger environment, there is optimization and scale that comes with specialization by role.

- **Business Owner or Stakeholder:** Drives the primary business objectives and funding for the solution. If the Elastic Stack environment is supporting multiple use cases, each business team might have its own business owner or key stakeholder.
- **Data Analyst:** Consumes the data insights provided by the Elastic Stack. The Elastic Certified Analyst certification is designed to prepare and validate these team members.
- **Ambassador:** Works directly with business teams for onboarding planning and initial data ingest into the Elastic Stack and helps them build dashboards or develop against the APIs.
- **Lead Architect:** Does design and capacity planning for the infrastructure, environment, and advisory functions for best use of the platform. We recommend that Lead Architects get certified to prepare them for questions of growth, best practices, scale, and performance.
- **Operations Lead and Operations Team:** Provisions the infrastructure and operations of the platform. This might be done by a DevOps team if your organization is also using the Elastic REST APIs or Elasticsearch Query DSL via a client library in your favorite language and automating via tools.
- **Search or Relevance Lead:** On Enterprise Search teams, responsible for optimizing search result quality and working with the business teams to ensure that user and business search expectations are being met.

The Elastic Services team offers a variety of services to assist our customers with knowledge and skill development. We focus on knowledge transfer and helping out customers become self-sufficient in Elastic technology. A few of our key Consulting capabilities are provided here:

Engagement Manager

Providing operational oversight on the project. Will assist with scheduling, resourcing and navigating your success plan

Technical Account Manager

Consultant with long-term advisory role

Embedded Architect

Long-term technical guidance and best practices

Embedded Engineer

Technical expert responsible for the hands-on delivery

Consulting Manager

Providing management oversight and engineer / specialist allocation

Our Elastic Training and Certification also helps to prepare your team for designing and operating the Elastic Stack. We have many courses that can help your teams — no matter their profile or familiarity with Elastic technology — become well-versed in Elastic products and solutions.

- Elastic training subscriptions provide access to our entire course catalog for a year — the best option for long-term learning. Standard and Professional subscriptions also include certification exams.
- Elasticsearch Engineer I and Elasticsearch Engineer II, designed for DevOps engineers and developers working on infrastructure or with APIs, prepare students for the Elastic Certified Engineer certification.
- Data Analysis with Kibana, designed for analysts and operators using Kibana for data visualization and dashboards, prepares students for the Elastic Certified Analyst certification.
- ECE Fundamentals is a free course that provides core training for installing and operating ECE.
- Our Private Elastic Endpoint Security course provides administrators and threat detection response teams with hands-on experience using the endpoint security capabilities of Elastic Security.
- Finally, there are a number of other free courses for new users to the Elastic Stack.

Additional references

Migrating from Splunk to the Elastic Stack: Data migration (blog)

Kibana for Splunk SPL Users (training)

In closing

The Elastic Stack provides speed, scale, and relevance via a native distributed architecture and simple design decisions that allow for future growth and flexibility. We hope you found this overview of design principles helpful in achieving a smooth migration so you can quickly get up and running with your new platform while avoiding the risk and cost of a complex migration.