The 5 Biggest Challenges of Elasticsearch

Understanding the Challenges of Elasticsearch and How to Overcome Them

What Is Elasticsearch?

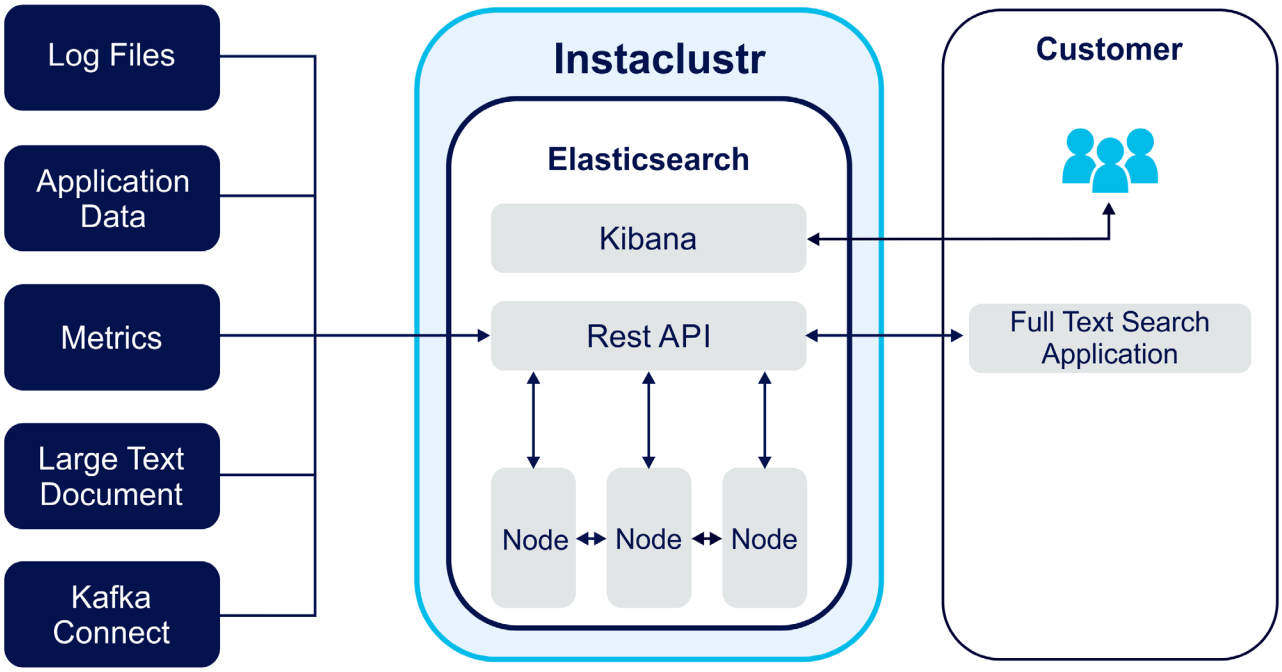
Before we dive into the major challenges of Elasticsearch, let's start with an overview of this powerful open source technology.

Elasticsearch is the most popular open source search and analytics engine in the world. The enterprise-proven solution brings unmatched speed, scalability, resilience, and flexibility to enable users to effectively and efficiently search across all types of documents and datasets.

As Elasticsearch is a document-oriented datastore, it requires no upfront schema definitions and is capable of storing, retrieving, and managing document-oriented or semi-structured data. Elasticsearch is Java-based and built on top of the Apache Lucene project, leveraging the Lucene Standard Analyzer to perform indexing, automatic-type guessing, and high-precision searches.

instaclustr

Elasticsearch offers REST APIs that allow easy integration into developers' existing technology stack. The Elasticsearch open source community has developed integrations to accelerate development across multiple programming languages, including Ruby, Java, PHP, Node.js, JavaScript, and others.



[Elasticsearch facilitating full text data search through a Rest API]

As a distributed RESTful search and analytics engine, Elasticsearch can divide indices into shards and create shard replicas. This allows the solution to parallel process high data volumes and rapidly deliver optimal query matches. From a performance perspective, Elasticsearch utilizes algorithms and data structures that are purpose-built to store and index data with maximum efficiency. Elasticsearch can also index and provide applications with freshly written data in real time, enabling many important security and analytics use cases (such as application monitoring or anomaly detection).

Elasticsearch features horizontal scaling that makes clusters simple to expand, and balances search and indexing automatically. Its built-in clustering support also makes it possible to run Elasticsearch across multiple servers within a single cluster, while always maintaining effective search performance.

Developers ready to empower their applications with robust and performant search functionality will find Elasticsearch a welcome addition to their technology stacks. The most common and beneficial use cases for Elasticsearch include:

- **Log analytics:** Elasticsearch facilitates analysis of unstructured or semi-structured logs, such as those produced by sensors, servers, websites, and other sources.

- **Security analytics:** Elasticsearch's ability to harness data in real time enables highly effective threat monitoring and responses to incidents as they happen. With Elasticsearch, events occurring across all systems and applications throughout an enterprise can be centralized for analysis. This can power holistic security solutions capable of instantly identifying and responding to attacks.

- **Clickstream analytics:** Elasticsearch makes it simple to interpret clickstream data and create reports that glean real-time insights into user behaviors and interactions with your web content. These reports can highlight successful approaches that drive traffic and interest in your enterprise and its offerings.

- **Full-text search:** Elasticsearch lends itself to creating powerful and personalized search experiences for an application's user base.

The Five Biggest Challenges of Elasticsearch

But Elasticsearch isn't without challenges, particularly if self-managing. In our time managing and providing support for open source technologies, we've observed the following to be some of the biggest challenges that operators encounter with Elasticsearch:

1. **Cluster size issues:** Elasticsearch features the scalability to enable it to go as big or small as you need. However, large deployments require thoughtful node and shard distributions to maintain performance and availability while handling whatever load is required. The solution is to leverage sharding and replicas to distribute indexes and promote performant response times.

2. **Too many shards:** With Elasticsearch, indexes are divided into physical spaces called shards. This lets you split data between hosts. But the catch is you have to define the number of shards at the time of index creation—and you cannot modify the shard allocation later without reindexing all of the source data. When you allocate shards, you should think about how you expect your dataset to grow over time. One of the common issues with Elasticsearch performance is allocating too many shards for small datasets. There is not a hard and fast rule for shard allocation, but generally it is common to see single shards can be between 20GB and 40GB in size.

3. **Crashes due to "mapping explosions":** Elasticsearch offers manual and dynamic mapping to determine how data is stored in indexes. There's a risk that if dynamic mapping isn't properly restricted, it can result in a mapping explosion that can cause Elasticsearch to crash. To avoid this risk, ensure that any dynamic mapping includes limits that keep it from getting out of hand.

4. **Crashes due to "combinatorial explosions":** If data is aggregated in a nested manner, bucket generation can become exponential and lead to a crash. Avoiding this issue requires careful attention to collection mode settings and control over how data is bucketed and collected.

5. **Bloated index templates:** Index templates are significant time savers, enabling rapid creation of powerful new indexes as needed. That said, a template with too much bloat will produce large mappings (see above), as well as lengthy update and debug completion times. It's best to simplify index templates by leveraging dynamic templates, or just keeping templates as lightweight as possible.

In each of these cases, support from experienced experts can make a big difference in maintaining efficient and issue-free Elasticsearch deployments.

The Advantages of Instaclustr Managed Elasticsearch and Support

Selecting a managed strategy for Elasticsearch removes the learning curve and challenges of implementing and integrating this solution. It also allows you to instantly realize the benefits of Elasticsearch so you can focus more time and resources on building and refining your own applications.

Instaclustr's managed solution for Elasticsearch provides 24x7 expert management from a team experienced in securely deploying, managing, operating, and scaling the solution to meet customers' specific real-time search, analysis, and data visualization needs. Instaclustr experts run Elasticsearch around the clock to perform log analysis from Elasticsearch clusters.

We are able to leverage highly refined operational disciplines developed through extensive experience in running and supporting Apache Cassandra and Apache Kafka deployments. Considering that enterprises utilize Cassandra to handle some of the world's most demanding applications, Instaclustr has developed rapid-response, expert support, and SLAs that now extend to our managed Elasticsearch customers. Instaclustr's experts have managed more than 100 million node hours of distributed data technology, and offer the

most reliable way to provision Elasticsearch in the cloud. Customers have the option of leveraging Instaclustr-managed Elasticsearch in their own cloud provider accounts or using an Instaclustr account for additional simplicity.

As part of Instaclustr's deep commitment to offering only non-commercialized open source data-layer technologies (that keeps customers in total control of their code and their solutions without expensive vendor lock-in), Instaclustr's managed Elasticsearch service is based on the Open Distro for Elasticsearch. The service is also SOC 2 certified, providing security and data protection equal to the highest industry standards. Daily scheduled data backups further secure Elasticsearch deployments and enable complete data restoration if a disaster affecting the customer's cluster does occur.

Instaclustr-managed Elasticsearch also features simple-to-use monitoring and provisioning APIs that customers can integrate into existing DevOps tools and processes. Instaclustr collects and monitors many different metrics for each node under management. By doing so, Instaclustr maintains constant and total oversight around the availability and performance of every cluster.

Instaclustr's Elasticsearch offering gives you a range of options and full control over how you wish to visualize and present data, including pairing Elasticsearch with Kibana to reap the benefits of improved analysis, visualization, and security features. Instaclustr also offers managed Elasticsearch integrations with the other advanced open source data-layer technologies that are part of the Instaclustr Managed Platform, including Apache Cassandra, Apache Kafka, Kafka Connect (which with the Elasticsearch Sink Connector lets you stream Kafka topics to your Elasticsearch cluster), and Apache Spark.

If you'd prefer to manage your own cluster but still get the benefit of our expert support team, you can also sign up for our Elasticsearch Support package. You'll enjoy 24x7x365 access to our team and guaranteed SLAs to ensure you have the support you need while managing Elasticsearch in-house.

You can get more information on Instaclustr's Elasticsearch solutions and sign up for a free trial **here**.

About Instaclustr

Instaclustr helps organizations deliver applications at scale through its managed platform for open source technologies such as **Apache Cassandra®**, **Apache Kafka®**, **Apache Spark™**, **Redis™**, **OpenSearch®**, and **PostgreSQL®**.

Instaclustr combines a complete data infrastructure environment with hands-on technology expertise to ensure ongoing performance and optimization. By removing the infrastructure complexity, we enable companies to focus internal development and operational resources on building cutting edge customer-facing applications at lower cost. Instaclustr customers include some of the largest and most innovative Fortune 500 companies.