A Brief Technical Review of Enterprise Search
Comparing ElasticSearch and AWS Kendra

Nicholas Bachman (bachman5)
CS 410: University of Illinois at Urbana-Champaign
October 24th, 2020

Intro

Everyone has become accustom to the tremendous benefits that Google search has brought into our everyday lives. We often take it for granted that a tiny search bar always brings us the topmost relevant results for any topic across the entire world wide web. Every day, Enterprise companies create large amounts of unstructed data in the form of Email, Power Point, Excel, Word, Wiki, pdf, etc. They often securely store this data in silos, behind firewalls, on network attached file systems (NFS/SMB) or in Content Management Systems (CMS) that are not accessible to the public internet. This is done for obvious security and compliance issue, but it cuts off access to popular public search engines like Google. Corporate citizens that need access to this private data become frustrated and unproductive because they are expecting a Google search experience for Enterprise data that often isn't available to them.

The McKinsey Global Institute (MGI) estimates that "Employees spend 1.8 hours every day—9.3 hours per week, on average—searching and gathering information. Assuming now an average yearly salary of $80,000, the inability to find and retrieve document costs organization, that employs 1000 workers, $ 25 million per year" (Noi, 2018). Enterprise search has become a ~$3B market opportunity because of these technical challenges and the business benefits that come from increased employee productivity. The scope of this technical review paper will compare the basic features of two commercially available Enterprise search offerings from Elastic and AWS Kendra.

Body

In 2004, Shay Banon was having challenges trying to create a cooking recipe application for his wife. He started by trying to use an open source search library called Apache Lucene to create a search box. It was difficult to use so he wrote his own software called Compass that interfaced with Lucene to make Java based indexing easier. Compass started becoming popular in the open source community and it evolved into what is now called Elasticsearch. Shay Banon's first github commit to Elasticserach was Jan 4th 2008. Over the next 12 years he and his open source community enhanced an entire ecosystem of Elasticsearch software that now includes Kibana, Beats and Logstash. Shay Banon is currently the CEO of publicly traded Elastic Inc (ESTC). (Kupfer et al., 2015)

"Elasticsearch provides near real-time search and analytics for all types of data. Whether you have structured or unstructured text, numerical data, or geospatial data, Elasticsearch can efficiently store and index it in a way that supports fast searches." (What is Elasticsearch, 2020). The Elastic Stack (ELK) expands features to collecting, aggregating and enriching data, which is out of scope for this review. Elasticsearch also has many additional use cases that are beyond Enterprise search.

One of the key benefits of Elasticsearch is that it can be installed and run anywhere. It supports traditional OS installs on MacOS, Windows, Debian, Ubuno, Red Hat, Centos and SLES. It also provides Docker container deployment options and integration with configuration management tools like Puppet,

Chef and Ansiable. This flexibility is especially valuable for large Enterprise customers that still have legacy internal system integration requirements such as Remedy, Confluence, JIRA, On-premise SharePoint or NAS. Most Enterprise customers are running in some form of Hybrid-Cloud architecture. It's best to run your Enterprise search application as close to the data sources as possible to maximize performance, minimize latency and mitigate security issues. Elasticsearch can accommodate deployments for both on-premise private clouds or in public clouds such as AWS, Google or Azure. This deployment flexibility is a major advantage when compared to SaaS offerings such as AWS Kendra.

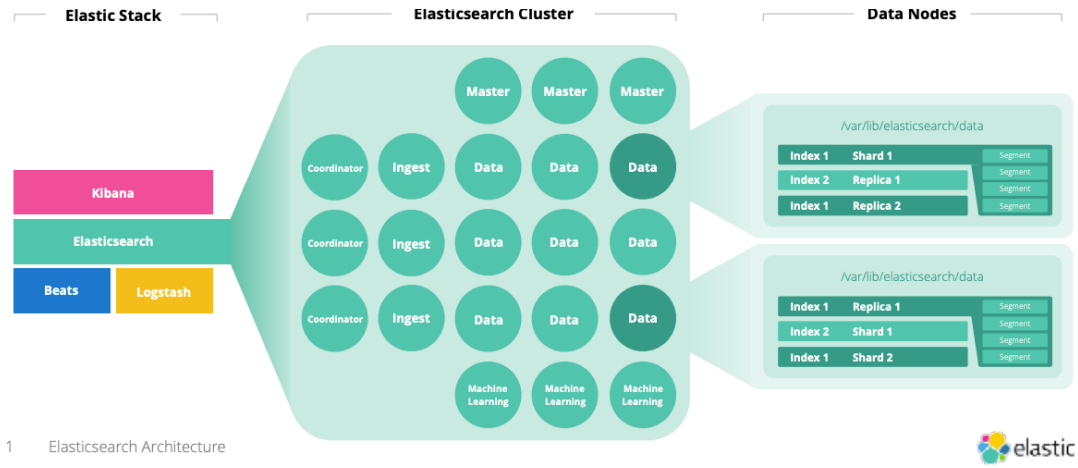

Figure 1. Elastic Architecture (Moore, 2020)

One of the key technical challenges with deploying Elasticsearch is building and scaling the infrastructure. Figure 1 shows the high-level architecture of an Elasticsearch implementation. There are Data, Master, Ingest, Machine Learning and Coordinator Nodes in the Elasticsearch Cluster. The amount of storage, memory, compute and network resources are dependent on each node type. For example, data nodes require extreme amounts of high-performance storage. Machine Learning Nodes require extreme amounts of memory and compute resources. The resource demand is also dependent on the application use case. Logging, metrics, security and APM use cases are index heavy. App search, site search and analytics are search heavy. (Moore, 2020). These underlying virtual resources can get expensive quickly in both public or private cloud Elasticsearch deployments if the environment isn't sized or scaled correctly. Self-Managed Elastic Stack is also licensed per data node with additional licenses for advanced machine learning nodes. Putting this responsibility of the infrastructure on the end user is one of the major disadvantages of Elasticsearch when compared to a SaaS offering such as AWS Kendra. This also led to Elastic providing their own managed service offering.

"Amazon Kendra is an intelligent search service powered by machine learning. Kendra reimagines enterprise search for your websites and applications so your employees and customers can easily find the content they are looking for, even when it's scattered across multiple locations and content repositories within your organization. Amazon Kendra is a fully managed service, so there are no servers to provision, and no machine learning models to build, train, or deploy." (AWS 2020)

Since Amazon Kendra is a SaaS offering, its key benefit is ease of use. Figure 2 shows the process of setting up Enterprise search to create an index, add data sources, test and deploy. Even non-technical

users can quickly and easily setup a search environment. AWS Kendra extracts away all the complexity of creating, deploying and maintaining the underlying software and hardware infrastructure. Regardless of the use case, the search infrastructure seamlessly scales up or down to match the requirements. AWS Kendra provides developers with the code for an existing React application to enable the main search page, search bar, results, facets and pagination.

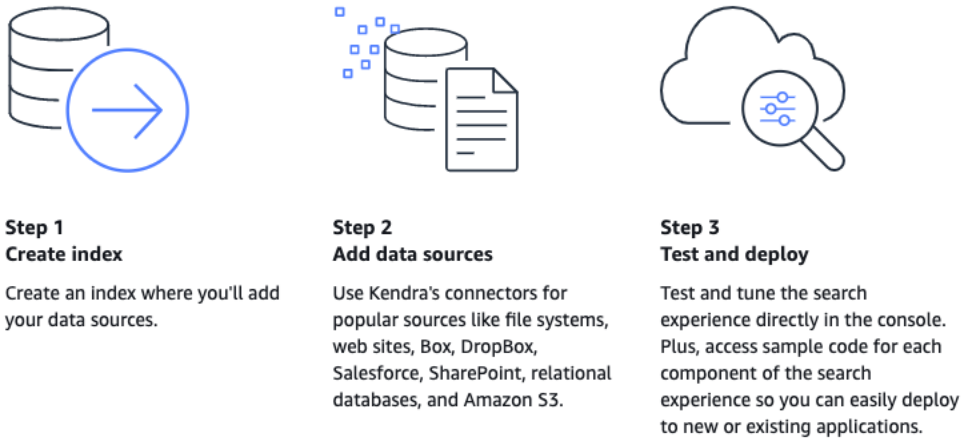

Figure 2. How AWS Kendra Works (AWS 2020)

The key technical challenge with AWS Kendra is flexibility. AWS Kendra is specifically targeted at AWS customers with the bulk of their data already in the public cloud. Since it's hosted in AWS, it can't run in a customer's data center or in a competitive public cloud provider. Kendra's connector integration works well for modern data sources like S3 or with other SaaS offerings. However, there is no easy way to connect Kendra to legacy private cloud applications or Content Management Systems (CMS). There is a Kendra SDK but no large open source community to support all the various integrations. Kendra currently only supports English which is also a serious limitation for large Global Enterprise customers.

Feature	AWS Kendra	Elasticsearch
Reading comprehension	No pre-training required. FAQ matching. English Only.	30 different language analyzers, including many languages with non-Latin character sets like Russian, Arabic, and Chinese
Document ranking	Deep learning based semantic search model without need for explicit configurations	Dynamic Mapping to index document. No Deep Learning or sematic level modeling
Connectors	S3, SharePoint, Salesforce, Servicenow, RDS databases, One Drive. Kendra SDK can be used to write custom connectors	Hundreds of community provided integrations, all common internal CMS, Outlook, JIRA, Confluence, etc.
Relevance / Confidence	Increase relevance for specific data sources, document freshness, view count or vote. Confidence scores are used to estimate accuracy of search results	By default, Elasticsearch uses BM25 for relevance. Dirichlet, Jelinek Mercer, DFI and IB can also be used and optimized.
Data Types	Unstructured and semi-structured data in .html, MS Office (.doc, .ppt), PDF, and text formats	Text, Shapes, Numbers, Vectors, Histogram, Date/time series, Flattened field, Geo-points/geo-shapes, Unstructured data (JSON), Structured data
Domain optimization	Deep learning models for domain optimizations and verticals. Example: HR, IT, Engineering, Healthcare, manufacturing	No features around domain optimization
Analytics	Not Available	Elastic Observability, Kibana and integrations with many BI Tools
Deployment	AWS Only	Local, Private Cloud, AWS, Azure, GCP
Security	Encryption of Data at Rest AWS or Customer CMK HTTPS/TSL	Encryption at rest and in transit, advanced role-based access control (RBAC) and attribute-based access control (ABAC)
Pricing	Developer Edition ($2.50/hour) Enterprise Edition ($7.00/hour) Connector Usage ($.35/hour)	Open Source Apache (Free) Gold ($x per node) Platinum ($x per node) Enterprise ($x per node)

Table 1. Feature Comparison of Elastic Search and AWS Kendra for Enterprise Search

Conclusion

The Enterprise Search market is an exciting space that will continue to see new innovations as machine learning applications mature. Elasticsearch has been at the forefront of search and analytics over the past decade. Since the Elastic stack is 100% open source, it will always outpace the feature delivery and integrations of SaaS offerings like AWS Kendra. AWS Kendra makes Enterprise search extremely easy to use and operate. Although it's not disclosed, the AWS Kendra service most likely uses many of the same open source software packages developed by the Elasticsearch community.

Enterprise companies that operate in a hybrid cloud architecture, maintain legacy CMS and have a deep bench of technical skills will see more benefits with Elastic. They will get the latest features sooner, have the ability to customize to a specific use case and integrate with applications outside of Enterprise search. The increased complexity is a trade off with a lower TCO if the Elastic environment is properly built, run and optimized as it grows.

Enterprise companies that are 100% hosted in AWS will see more benefits with AWS Kendra. Companies without a deep bench of technical resources will enjoy the simplicity of Kendra. Web developers will have a simple way to add cutting edge Enterprise search capabilities to their applications. Simplicity does however come at a cost. The pay per hour of use module with added cost of connector usage may lead to a higher TCO as the environment grows.

References

Noi, D. D. (2018). Do workers still waste time searching for information? https://blog.xenit.eu/blog/do-workers-still-waste-time-searching-for-information#:~:text=According%20to%20Interact%20Source%2C%2019.8,the%20workday%2C%20searching%20for%20information.

Kupfer, D., Author Diana Kupfer @dianakupfer All Posts by Diana Kupfer Working at S&S Media since 2011, Diana Kupfer @dianakupfer All Posts by Diana Kupfer, Kupfer, A., & Working at S&S Media since 2011. (2015, September 15). A brief history of Elasticsearch. Retrieved October 24, 2020, from https://jaxenter.com/elasticsearch-founder-interview-112677.html

What is Elasticsearch? Elastic. https://www.elastic.co/guide/en/elasticsearch/reference/current/elasticsearch-intro.html

Moore, D. (2019, November). Quantitative Cluster Sizing. Retrieved October 24, 2020, from https://www.elastic.co/webinars/elasticsearch-sizing-and-capacity-planning

Elastic Corp (2020). Elastic Stack features. Retrieved October 24, 2020, from https://www.elastic.co/elastic-stack/features

Amazon Web Services. (2020). Kendra. Retrieved October 24, 2020, from https://docs.aws.amazon.com/kendra/latest/dg/what-is-kendra.html