

What does Elastic do?

Yet another database! feat. search



Justin ✓

🔒 Mar 1, 2022

♡ 19

💬 3



Apologies for the delay on this one. I was going to publish last week but felt given the circumstances it was better to wait it out a bit. We've got some really exciting stuff coming over the next couple of months, including a guide for how to get more technical, along with posts about WebAssembly, GraphQL, Data Lakes, and ORMs.

The TL;DR

[Elastic](#) is the commercial company behind Elasticsearch, a popular open source database for storing and searching unstructured data.

- Companies collect loads of **unstructured data** in the form of logs, requests, sessions, server metrics, etc.
- Elasticsearch is a **database** to store that data, and a **search engine** to easily comb through it
- Unlike MongoDB or MySQL, Elasticsearch is an **analytical database**, not a production one
- Elasticsearch is commonly used with **Kibana**, its sister **data visualization tool**

Elasticsearch is a highly popular option for use cases around log management, typically for larger companies. The company behind the open source software [went public](#) back in 2018 and did around \$400M in revenue in 2020.

Another database? Some taxonomy

Yet another database!? Yes, my dear readers, another database. But Elasticsearch isn't like other databases; it's **use case specific**, meaning it was designed for doing specific things with particular types of data. One of its flagship features is also **built-in search** (hence the name), which is now

becoming common in the NoSQL database world, but was novel when it first released. To understand *any database*, you first need to understand why teams use it, and it's there we begin this installment of Technically.

→ **OLTP vs. OLAP databases**

Broadly speaking, there are two types of databases out there.

(1) The first category is used to **power the apps that we know and love**: they store information about us, our profiles, and any content related to us, like our Tweets on Twitter or our emails on Gmail. These are known as OLTP databases – an acronym for OnLine Transactional Processing – and they're optimized for many small queries in quick succession with few joins. MySQL, PostgreSQL, Redis, and MongoDB are all (primarily) OLTP databases.

(2) The second category is used to **store long term data and analyze it**. That analysis can be business related – like wondering what revenue is this month – or operational, like figuring out which [Kubernetes node](#) is causing the app to keep crashing today. These are known as OLAP databases – an acronym for OnLine Analytical Processing – and they're optimized for fewer, more complex queries with many joins. Snowflake, BigQuery, and Elasticsearch are all OLAP databases.

Elasticsearch fits into this latter category. Companies typically don't use Elasticsearch as their primary data store backing their apps. It won't store user information or anything mission critical to the actual app the company sells. It usually won't interact with your web app directly. Instead, it's primarily for storing **performance-related data** and **analyzing it down the road**.

Deeper Look

While Elasticsearch is *primarily* used as an OLAP database, some teams do use it to power [user facing search experiences](#) (think: searching your emails or past tweets). This use case is somewhere in between OLTP and OLAP.

Deeper Look

→ **Structured vs. unstructured data**

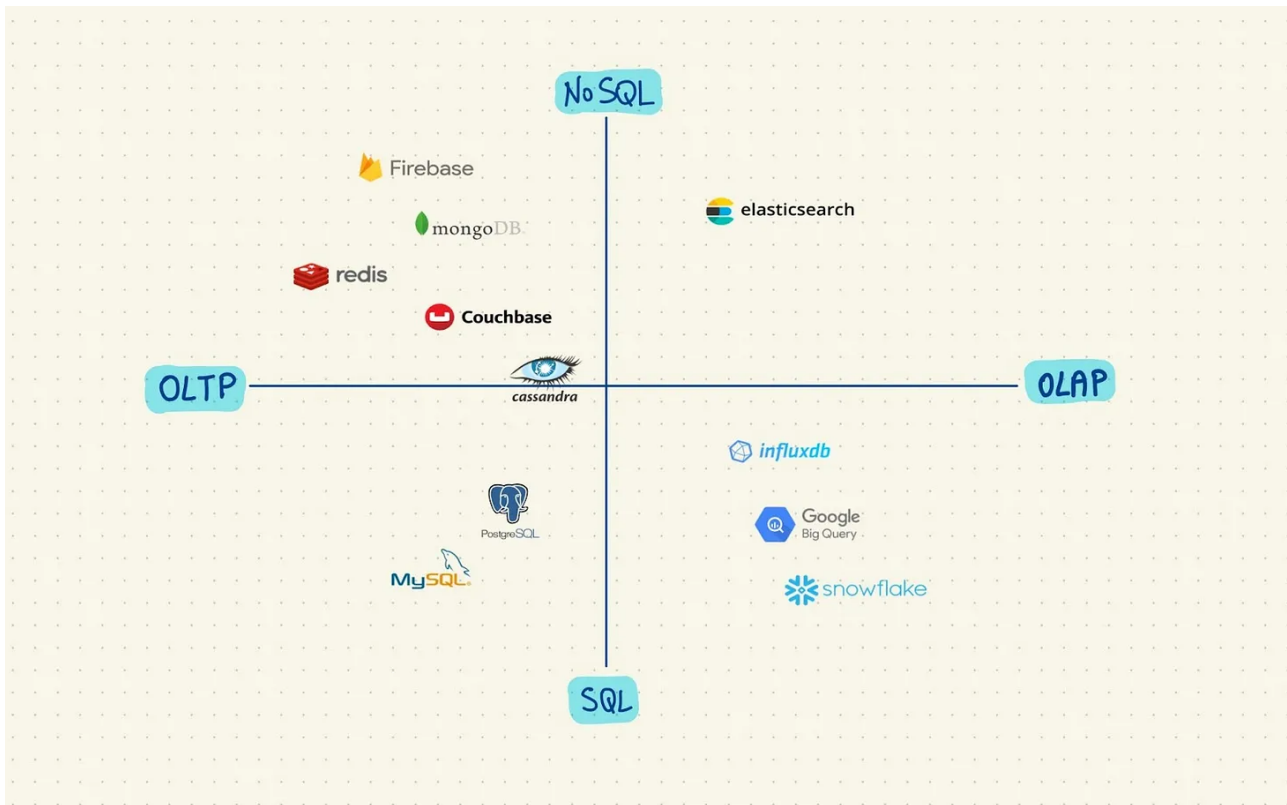
Data usually comes in two forms: **structured and unstructured**. Structured data is organized into familiar table structures, like you'd see in Excel, while unstructured can just be giant blobs of text or other similar data. A user in your production database is structured:

user_id	first_name	last_name	created_at
0bdsaouDg_audi2er	Duncan	Idaho	'1994-01-13'

While a log that your server sent when there was an error can be unstructured, or even just a bunch of loose text:

```
[kafka.log][INFO] Retrying leaderEpoch request for partition logs-0  
as the header reported an error: NOT_LEADER_FOR_PARTITION
```

Generally, SQL databases like MySQL or Snowflake are best for storing structured data (be it transactional or analytical), while NoSQL databases like MongoDB or Redis are best for storing unstructured data. Elasticsearch is an unstructured data store.



So with the above, admittedly rudimentary taxonomy in mind, and the understanding that Elasticsearch is an analytical database used for unstructured data, we can dive into what teams actually *use it for*.

Elasticsearch's primary use cases

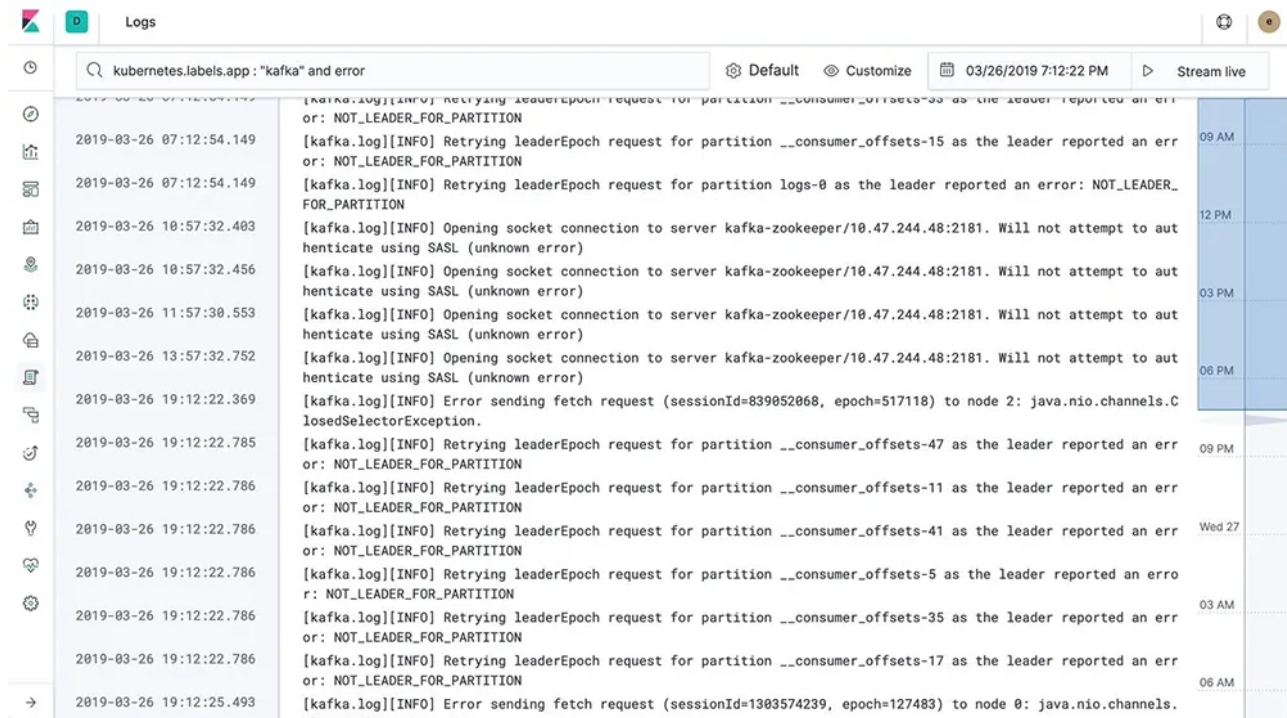
Elasticsearch's primary use cases revolve around things that commonly need, you guessed it, search. One big theme centers around infrastructure management, but teams also use it for security and even user facing search engines.

1) Application and server logs

Your application – as well as the server(s) that you run it on – shoot out tons and tons of logs. They're basically just text that say what's going on, like "we're installing this" or "this thing failed" or "we're starting up this program." But in those logs, my friend, is gold. When things go wrong and you're trying to figure out why, these are some of the first things engineers will turn to.

With Elasticsearch, you can stream your logs from the source, store them indefinitely (or set a retention window), and search them granularly. In this

screenshot, the user is searching for logs that relate to their [Kafka cluster](#) with the word “error” in them.



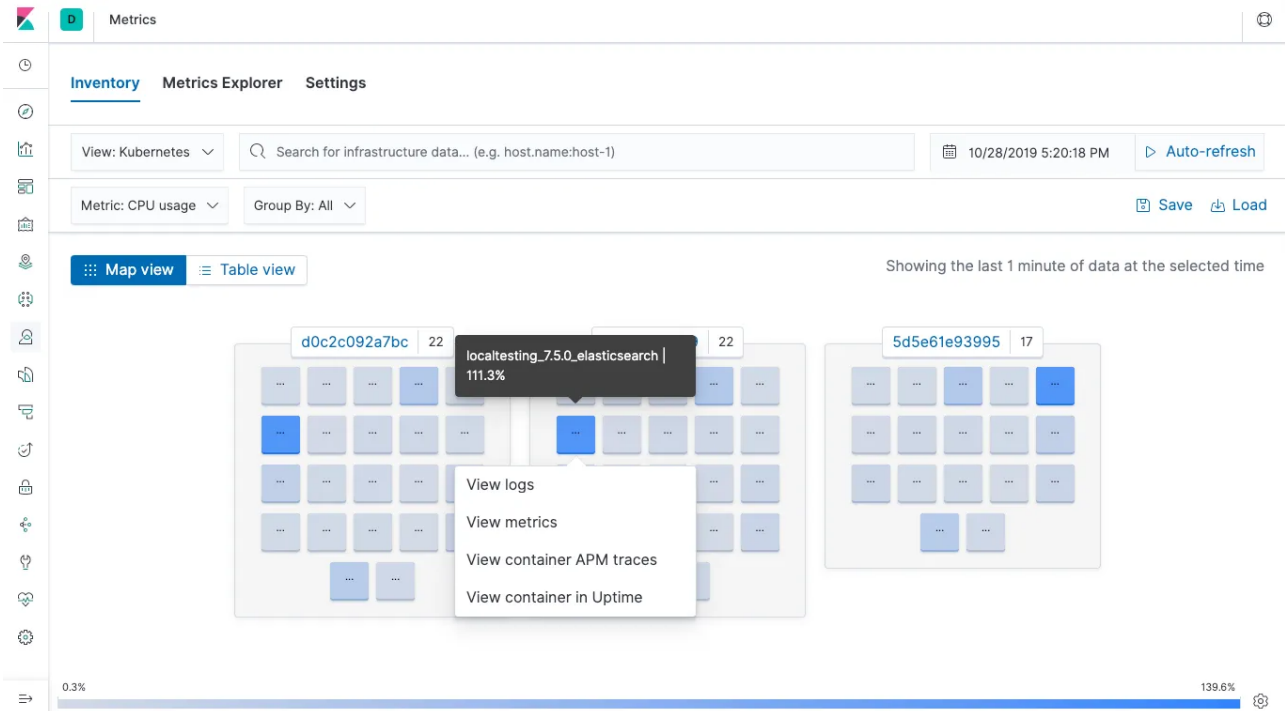
Confusion Alert

Note that the screenshots in this section are actually of Kibana, not Elasticsearch. You can think of Elasticsearch as the backend that stores the data and powers search, while Kibana is the frontend for visualization and UI / filtering. They are almost always a package deal.

Confusion Alert

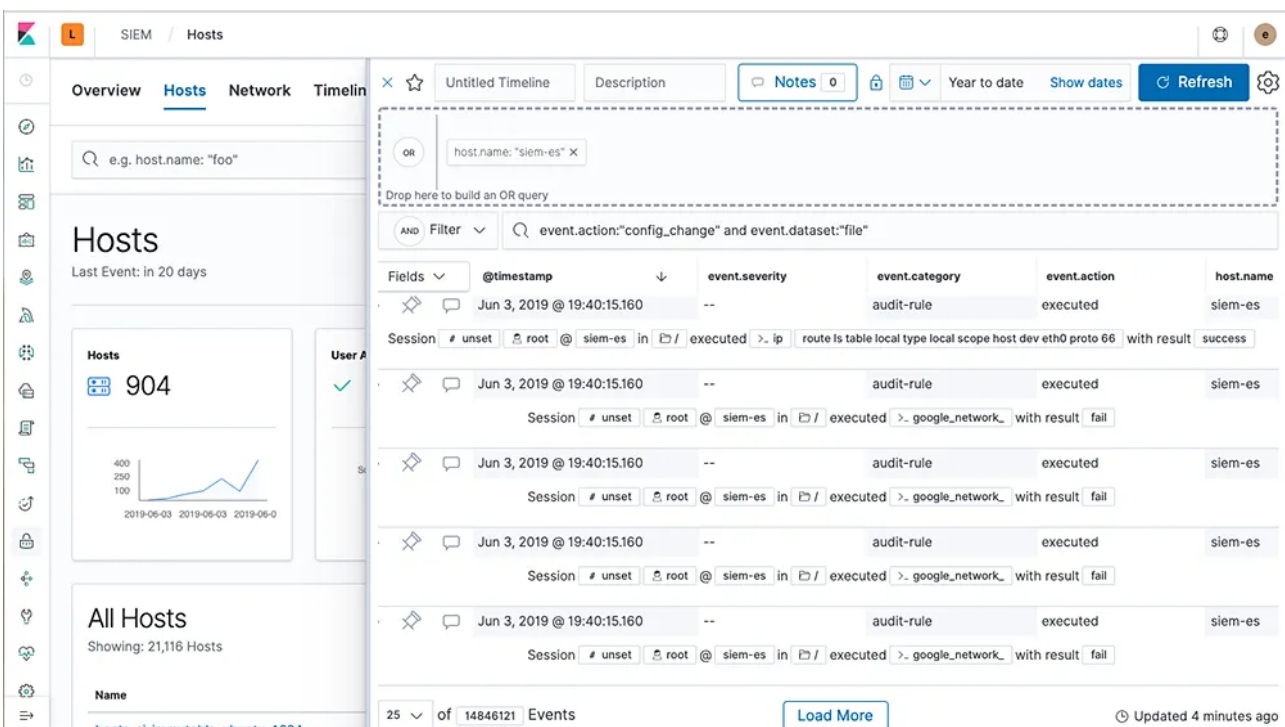
2) Infrastructure metrics

If you're running your application on complex infrastructure like [Kubernetes](#), or really any distributed system, developers will collect metrics on the performance of each individual element in that system like CPU usage or utilization. Elasticsearch allows you to store that data and use Kibana to visualize it hierarchically.



3) SIEM

SIEM stands for Security Information and Event Management, and it's the practice of documenting and analyzing any access to your internal systems. At larger organizations, IT admins look at logins from external computers and other data points to find patterns and prevent breaches. You can store this type of data in Elasticsearch and easily visualize and filter it in Kibana:



These are 3 primary examples, but there's so much more you can do with Elasticsearch. Elastic has a [great post on their blog](#) explaining some of the cool stuff their users do with the product.

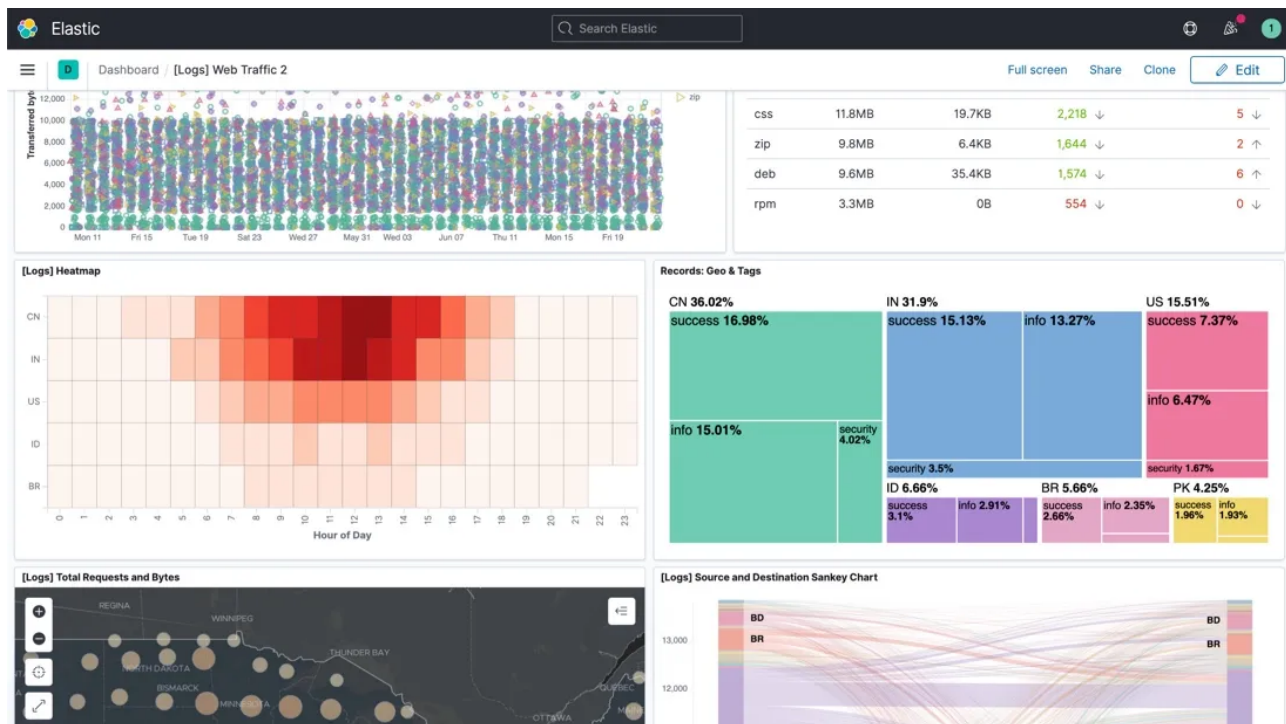
The ELK Stack and Elastic's business model

→ The Elastic or ELK stack

While Elasticsearch is the database and search engine here, it's meant to be used as part of a **stack** of other related, open source tools. Elastic calls it the [ELK Stack](#) which is...a name. Elastic says it stands for **Elasticsearch, Kibana, Beats, and Logstash**, or in other words their acronym department is out to lunch? Anyway, it's worth understanding the 3 tools in there that aren't Elasticsearch:

1. Kibana

[Kibana](#) is an open source visualization tool meant to be used on top of Elasticsearch. It's really most accurately described as the frontend for Elasticsearch, and makes it much more interactive. You can search and filter in the UI, build visualizations and graphs, and even do Machine Learning like anomaly detection. Most everyone using Elasticsearch is also using Kibana.



2. LogStash

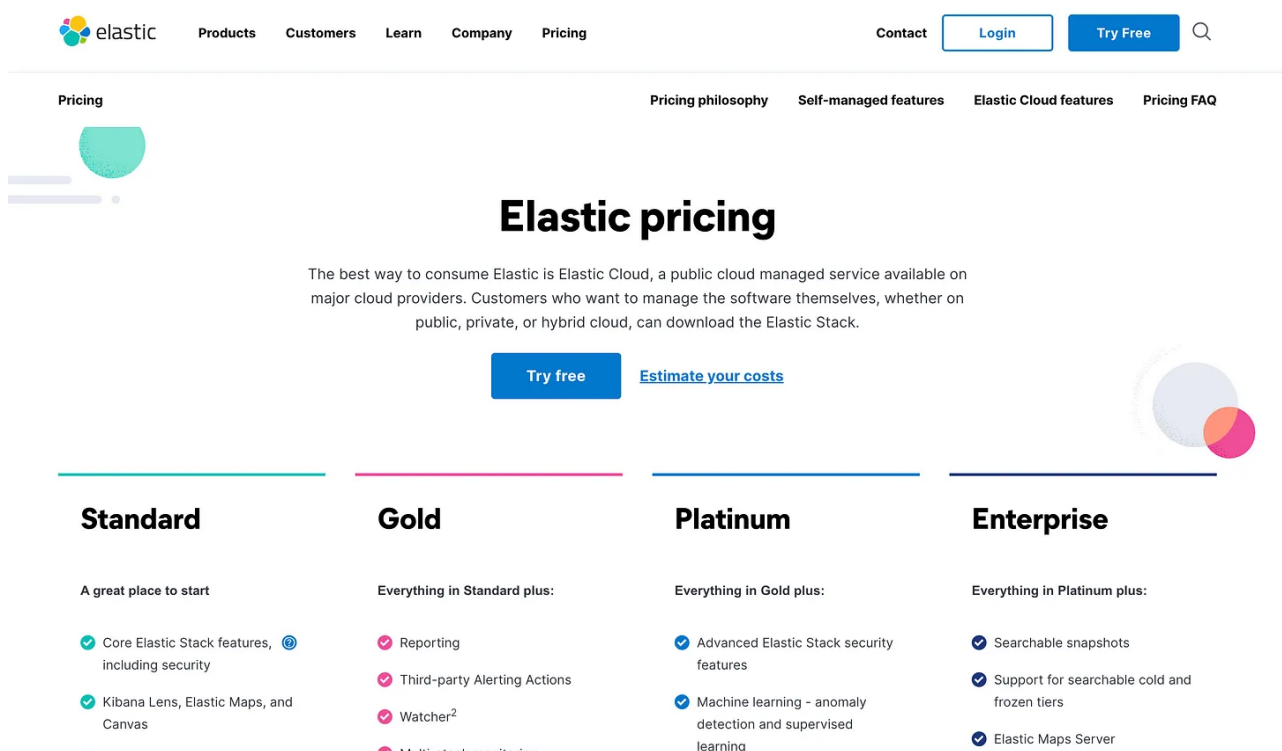
[LogStash](#) is a utility for helping you actually *get* your data into Elasticsearch. You install it on the servers from which you desire to send data from, and it helps you collect that data, transform it, and then send it to your Elasticsearch cluster. It has pre-built connectors for common log formats and can infer structure that helps you store this stuff in Elasticsearch properly. No screenshots here, as it's a [headless tool](#).

3. Beats (not by Dre)

You can think of [Beats](#) as a lightweight version of Logstash. [It does the same thing](#) – help send data from your sources to Elasticsearch – but it's a series of smaller, single purpose utilities instead of a bigger general purpose one. The smaller size and ease of use can be beneficial if the data you're sending to ES is simple.

→ Elastic's business model

All of the ELK stack products are open source and free, if you so desire to run them on your own infrastructure. Elastic makes money in the same way as most commercial open source vendors like [Confluent](#) or [MongoDB](#) – by charging you to abstract the infrastructure and run it for you. There's a lot of configuration that goes into self hosting this technology and [fine tuning it to your use cases](#), and a lot of teams don't want the hassle.



Elastic pricing

The best way to consume Elastic is Elastic Cloud, a public cloud managed service available on major cloud providers. Customers who want to manage the software themselves, whether on public, private, or hybrid cloud, can download the Elastic Stack.

[Try free](#) [Estimate your costs](#)

Standard	Gold	Platinum	Enterprise
A great place to start	Everything in Standard plus:	Everything in Gold plus:	Everything in Platinum plus:
<ul style="list-style-type: none"> Core Elastic Stack features, including security Kibana Lens, Elastic Maps, and Canvas 	<ul style="list-style-type: none"> Reporting Third-party Alerting Actions Watcher² Multi-stack monitoring 	<ul style="list-style-type: none"> Advanced Elastic Stack security features Machine learning - anomaly detection and supervised learning 	<ul style="list-style-type: none"> Searchable snapshots Support for searchable cold and frozen tiers Elastic Maps Server

Using Elastic's cloud product, you pay for the infrastructure you use (more money for larger VMs, etc.). They have a few [tiers of plans](#) that start at \$20/mo or so and go way up to hundreds of thousands of dollars. Elastic cloud only accounted for [22% of revenue in 2020](#) though – the majority of Elastic's revenue comes from large, recurring enterprise contracts. For those, Elastic will deploy these products on your own infrastructure or do a [hybrid managed cloud situation](#).

Questions? Thoughts? Leave a comment right here.

3 Comments



Write a comment...



Jack Mar 7, 2022

Are you planning on doing a company breakdown of Splunk in the near future?

♡ 2 Reply Gift a subscription Collapse ...



Karl Mar 1, 2022

We are a market data company and use elasticsearch to power our user facing search. Can you ELI5 why we would do that, what the pros and cons are, and what the alternatives would be?

♡ 1 Reply Collapse ...

1 more comment...
