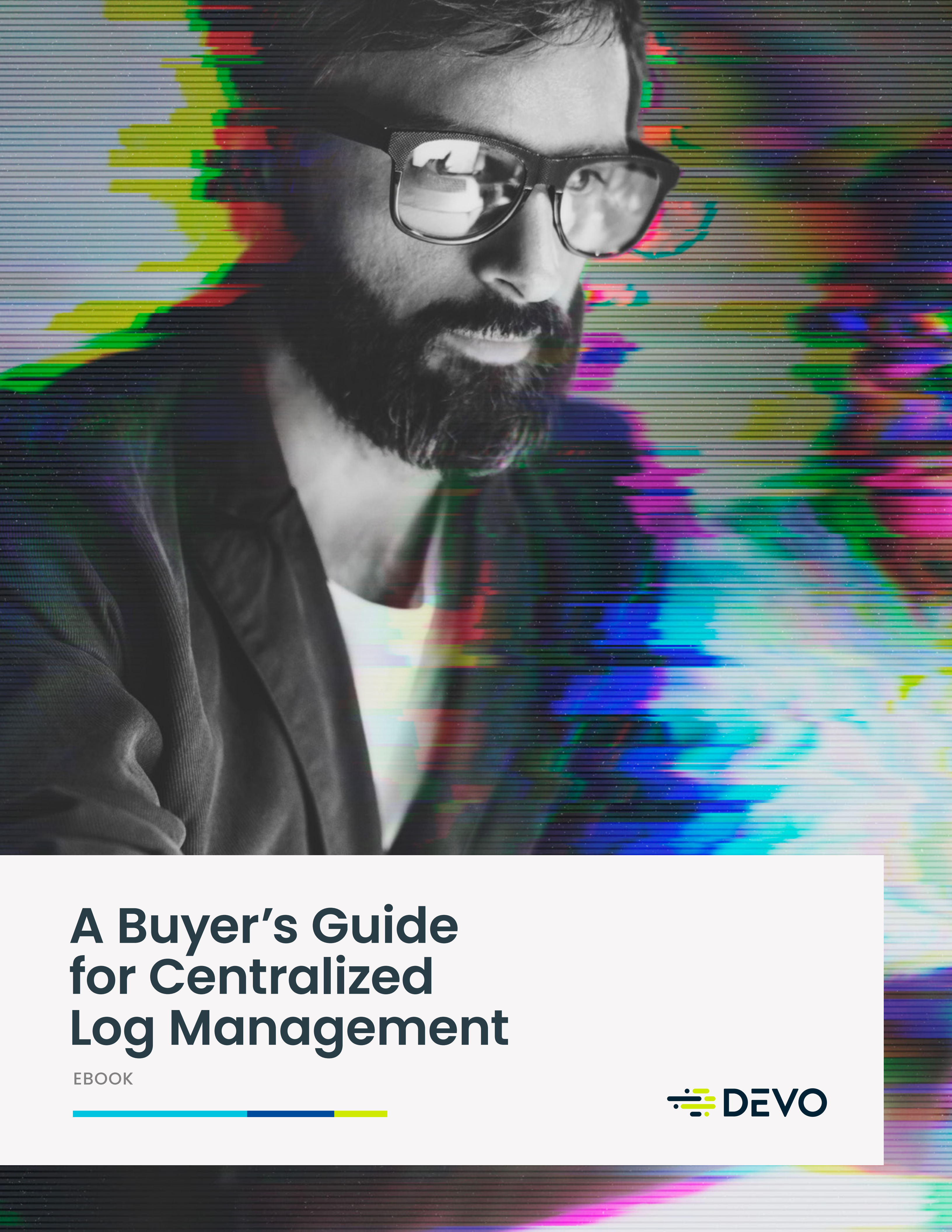


A Buyer's Guide for Centralized Log Management

EBOOK

DEVO

Introduction

When evaluating centralized log management solutions, it is important to evaluate deployment types, architecture, features, and cost models.

Elastic, Splunk, and Devo are all considered market-leading centralized log management solutions. All three also have SIEM components. So, how do the three stack up against each other? Which one should you choose? The best way to get a true apples-to-apples comparison of cost, functionality, and performance is to evaluate each vendor's SaaS offering.

That's because the SaaS deployment model is emerging as the clear choice for centralized log management. But since this is a rather recent market trend, it is important to review more traditional on-premises deployment models for centralized log management to learn why SaaS is emerging as the clear winner. In addition to evaluating SaaS offerings, we also will look at more traditional deployment options like deploying off-the-shelf software in your own self-managed cloud or building it yourself using open-source software.

BUILD IT YOURSELF—GOING THE OPEN-SOURCE ROUTE

For those who choose to build it themselves, the clear choice is Elastic (the ELK stack). It's open-source, well documented, and doesn't have a big upfront licensing cost—it's "free." Most DevOps teams have a small ELK stack running in their Dev/QA environment for testing and debugging purposes already—extending Elastic to production for all logs can make sense.

It is important to recognize that open source is not really "free" and comes with several not so hidden costs. Even mid-sized Elastic deployments require a significant capex spend on infrastructure to support its large compute and storage requirements. It also requires significant operational resources to manage and maintain all that infrastructure along with the Elastic deployment itself. There are a few key things to keep in mind if you want to build it yourself:

- **Sizing is critical:** Search performance can be slow depending on infrastructure sizing and data ingest rate.
- **Historical data needs:** Storage limitations make it costly to have more than 30 days of historical data available to search.
- **DR or no DR:** creating a disaster recovery deployment for Elastic doubles the cost and operational effort.
- **Growth requires re-architecting** and every 3 or 4 years the demands of the Elastic deployment can outgrow the capabilities of the hardware on which it is deployed. This requires re-architecting and re-deployment on newer, bigger, and more expensive hardware.

BUY IT

For organizations without the in-house expertise to build a centralized log management solution, one option is to buy it. Almost immediately after launching in 2003, Splunk became the top off-the-shelf option for centralized log management. Since Splunk deployed easily "out of the box," buyers could get it up and running more quickly than an Elastic deployment. And as a for-profit company, Splunk could develop new features that customers wanted much faster than Elastic.

Splunk also quickly built an army of tech support staff, consultants, and partners to provide the help and expertise customers needed. These capabilities enabled Splunk to become the market leader in centralized log management. But buyers soon realized that running Splunk was not without challenges. Like Elastic, Splunk deployments required a significant capex spend on hardware to support its massive compute, memory, and storage needs. And while Splunk support was generally perceived as good, it still required sizeable operational resources to deploy and maintain it. Finally, as Splunk's popularity grew, its licensing model became more aggressive. This led to price increases, forcing many organizations to reevaluate the total cost of ownership of their Splunk deployment.

Buyers who don't want to run Elastic or Splunk on-prem can purchase a software license and deploy it in the cloud.

BUILD IT IN THE CLOUD

Buyers who don't want to run Elastic or Splunk on-prem can purchase a software license and deploy it in the cloud. Deploying Elastic or Splunk in the cloud (usually AWS) delivers all the benefits of the solution, but without the big capex spend that comes with on-prem deployments. It also gives buyers the ability to monitor their cloud environments the same way they monitored on-prem environments.

This option initially worked well, as customers took their first small steps toward moving applications and services to the cloud. However, as the number of applications and services running in the cloud grew sharply, these self-managed deployments became more expensive and difficult to maintain. This deployment method still requires operational resources to manage, only now the teams doing the work needed to be cloud experts. Also, all the architectural limitations of these legacy solutions became an issue, the only difference being they are running on someone else's infrastructure. And the cloud infrastructure costs are usually very difficult to predict and budget, leading to unpleasant cost surprises. Buyers who take this route simply trade managing a costly, complex on-prem infrastructure to managing a costly, complex cloud infrastructure.

The SaaS model offers the shortest time to value and thus the fastest ROI compared to the other models.

SaaS

All of the above brings us to today's era of software-as-a-service (SaaS) delivery of centralized log management. This model requires none of the capex spend of the "Build It" and "Buy It" models. It also has none of the operational overhead of the "build it in the cloud" model. And since cloud infrastructure costs are usually included in the pricing, it is simple and easy to predict annual costs year after year. The SaaS model offers the shortest time to value and thus the fastest ROI compared to the other models.

Devo, a next-generation centralized log management vendor, architected its solution as a SaaS offering to take advantage of the cloud's many benefits. Splunk and Elastic are working to catch up to this trend by offering SaaS deployment options to capitalize on the popularity of this model. But buyers should characterize the operational requirements and additional costs of these SaaS variants, as support may be required to carry out common management tasks and they have additional fees for key functionality.

COMPARING SPLUNK, ELASTIC, AND DEVO

To compare these three solutions, let's compare each of these SaaS offerings across 3 dimensions: architectures, features, and costs. To determine who is better, we'll look at the advantages and disadvantages of each vendor in each category and see who comes out on top.

#1

Architecture Comparison Between Elastic, Splunk, and Devo

ELASTIC ARCHITECTURE

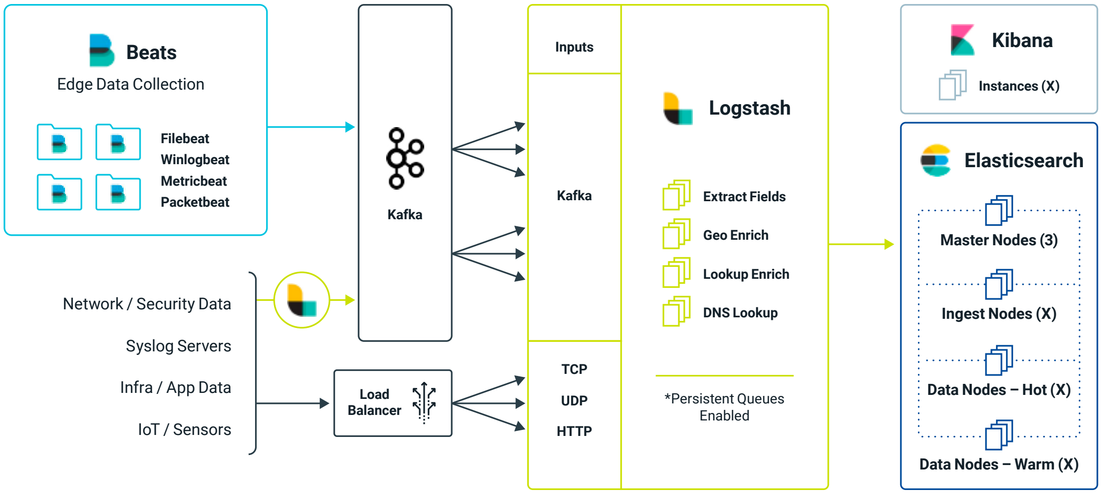
The Elastic architecture uses a variety of methods (Beats, syslog, etc.) to forward data to Logstash, which processes the data and sends it to the Elasticsearch tier (based on Apache Lucene) for indexing and storage. Kibana is used for visualization. In production installations, Kafka is used as the message broker between Beats and Logstash, and to ensure incoming data from Beats is not lost. Zookeeper is used for Kafka configuration and synchronization. A diagram of these components is shown below.

One advantage of running Elastic in a SaaS deployment is having the complex work of deploying and managing all the components handled as part of the service. This alleviates a significant amount of operational pain for the customer, enabling them to focus on leveraging the data. But this approach is not without its downside.

The disadvantages of a SaaS deployment of this complex architecture involve scalability, performance, and cost. Because Elastic in the cloud is a lift-and-shift of its on-prem architecture, all the same challenges of on-prem deployment carry over to the cloud. Since data must be indexed and parsed before it is searchable, there can be a significant lag from when the data is ingested to when it is available for search. This problem becomes more acute during big bursts of data on the ingest side. The lag can be especially problematic when analysts are investigating high-priority operational or security issues. Because data must be parsed prior to ingest, its format must be known. If the format changes, the data must be reindexed, a lengthy process that impacts search.

Another capability to benchmark is search performance. When an index becomes large (>50GB), Elastic splits it into "shards." Searching data across shards can be slow, and performance degrades as more data is ingested. Finally, you must consider the storage footprint. Depending on how you decide to index data (truncate original data or not), your ratio of ingested data to index size can be anywhere from .5x to 1.4x. This results in large storage requirements for hot data. For this reason, buyers typically opt to keep hot data for only about 30 days. It is possible to search against a warm data tier, but performance is not the same. Inconsistent search speeds due to the way data is tiered and sharded can result in slow search performance, especially for historical data. This can impact the speed of threat investigations to find the first instance of an IOC. Similarly, the rate at which threat hunting queries can be answered is compromised when querying across large data sets.

> One advantage of running Elastic in a SaaS deployment is having the complex work of deploying and managing all the components handled as part of the service.



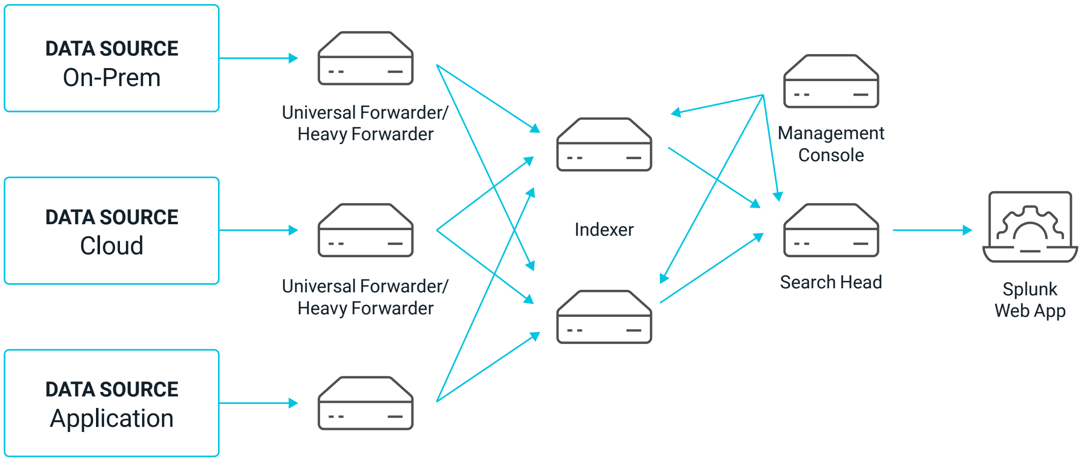
Like Elastic, Splunk was designed for on-premises deployment, so its cloud offering also is a lift-and-shift version of the original.

SPLUNK ARCHITECTURE

The Splunk architecture is similar to Elastic's, but it is slightly simpler. Data sources send information to Splunk's heavy forwarder via Rsyslog, NXLog, etc. The heavy forwarder pre-filters the data before sending it to the Splunk indexer. Search heads distribute searches to one or more indexers and search results are available directly from the browser via Splunk's web interface. A simple version of this architecture is shown in the below diagram.

Like Elastic, Splunk was designed for on-premises deployment, so its cloud offering also is a lift-and-shift version of the original. The main advantage of Splunk's architecture compared to Elastic's is simplicity. Although they are similar, Splunk has fewer moving parts, which results in fewer cloud instances to support. But because of the similarities, many of the same drawbacks exist. Since data must be indexed before it can be searched, there are potential delays to queries, especially during bursts of data. When data models are used, there can be up to 15 minutes of lag between data ingest and searchability.

Each indexer can't ingest much more than 250GB a day, which necessitates a large number of indexers at scale. This significantly increases cloud infrastructure costs, which Splunk passes on to the customer. Lastly, each of Splunk's add-on applications (IT Service Insights, Enterprise Security) requires additional data models to run against the indexers, further slowing search performance. This diminished search performance affects everything from single user queries to dashboard refreshes.



Of the three vendors in this comparison, Devo's solution is the only one with an architecture that is completely cloud-native.

DEVO ARCHITECTURE

The Devo architecture provides a unique method of ingesting and storing data. This enables significant performance benefits in ingest rate, search performance, and data compression. Devo uses a component known as a Relay to aggregate data from sources, tag it, and transmit it via a secure connection. Data from one or more Relays enters an event load balancer before being sent into the Devo Data Node for storage. As Data Nodes scale up, Meta Nodes coordinate searches across multiple Data Nodes. This architecture scales horizontally, as shown in the following diagram.

The tagging of data by the Relay tells the Data Node how to store it, as in, for example, web.apache.error. The tags can have many layers to further identify and segment the data. Tagged data is stored in a hierarchical file system in the Data Node. This eliminates the need for indexing data on ingest. Storing data in a hierarchical file structure also makes data compression much more efficient, since today's compression algorithms are very effective.

This architecture offers several advantages over legacy architectures such as Splunk and Elastic. First, data does not need to be indexed on ingest, which means it is immediately searchable. This also enables real-time alerting.

Second, the lack of indexing on ingestion means Devo scales very efficiently. A single Data Node can ingest 2TB of data per day. This means Devo requires significantly less cloud infrastructure to scale up, compared to Elastic and Splunk, which directly translates to cost savings for your organization.

Third, since data is parsed at query time, not on ingest, it is always stored raw and never changed. Data format changes have no impact on ingestion and search. Data never requires reindexing if format or source changes. This makes the Devo architecture more tolerant to change, compared to Elastic.

Finally, Devo's unique method of storing data in hierarchical flat files leads to an average 10:1 compression ratio of data ingested vs. storage size. This results in Devo keeping all data hot and searchable for 400 days, which is included in the standard pricing. This provides you with substantially more hot searchable data than Elastic or Splunk can affordably provide.

ARCHITECTURE COMPARISON SUMMARY

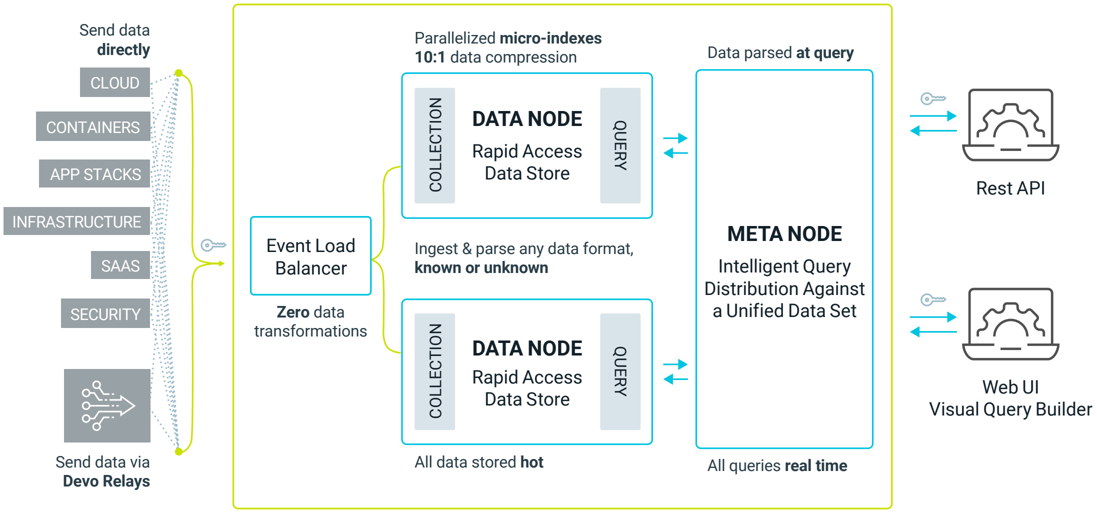
In reviewing the three vendors, it becomes clear that Elastic offers the least scalable and most complex architecture. Using Elastic as a SaaS solution eliminates much of the operational pain of maintaining the solution. However, it does not mitigate the lack of real-time search and alerts, the impact of scalability on slow search performance, limited hot data storage, and high cloud infrastructure costs due to its complex architecture.

Splunk is slightly better than Elastic but suffers many of the same problems including the lack of real-time search and alerts, slower search performance, and inefficient storage that results in a lack of hot historical data.

The Devo architecture offers real-time search and alerts, scalability, search performance, and 400 days of always-hot, searchable data for historical investigations.

The Devo architecture offers real-time search and alerts, scalability, search performance, and 400 days of always-hot, searchable data for historical investigations.

DEVO NO-COMPROMISE ARCHITECTURE



#2 Comparing the Features of Elastic, Splunk, and Devo

ELASTIC FEATURES

As an open-source product, Elastic provides a rich feature set right out of the box. You can use Elastic not just for log data, but also for SIEM, ITOps, and APM use cases. Infrastructure metrics, such as CPU and memory utilization, can be combined with logs to troubleshoot infrastructure. Elastic can import data from a distributed tracing system such as Zipkin to help troubleshoot slow application performance. And Elastic has a SIEM module that includes a detection engine, threat hunt capability, case management functions, and some basic endpoint security. Elastic also has ML algorithms that spot anomalous behavior or activity to aid in detections. With the exception of endpoint security—which is only available with Elastic's Enterprise-level subscription—all of Elastic's features are out of the box.

Elastic does not include a SOAR (security orchestration and automated response) platform as part of its solution but does offer tight integration with IBM's Resilient SOAR product.

SPLUNK FEATURES

Splunk takes a modular approach to functionality. If you want the ability to perform infrastructure monitoring, you have to pay extra for the Splunk ITSI (IT Service Intelligence) premium app. The same is true of Splunk Enterprise Security. The end result is Splunk's cloud offering includes the same features as the Elastic Cloud offering, but you need to pay more to access them.

Until October 2019, Splunk did not include distributed tracing as part of ITSI, but its acquisition of SignalFx rectified this. Splunk does have a SOAR platform (Phantom), but it is not included in Splunk Cloud. Users must run Phantom in their own AWS, GCP, or Azure cloud environment and integrate it with Splunk Cloud.

FEATURE COMPARISON SUMMARY

To make comparisons clear, this table tallies all the features available in each vendor's SaaS offering.

There are many similarities among the three vendors. Devo's ability to perform real-time search and its inclusion of 400 days of hot, searchable data give it a slight edge over Elastic and Splunk. Those features make Devo powerful for troubleshooting real-time problems as well as conducting historical investigations with speed and accuracy.

DEVO FEATURES

Unlike Splunk, Devo was never a modular product. Customers have always received full functionality out of the box. Since its debut, Devo has provided customers with full centralized log management functionality along with ITOps, APM, and SIEM functionality. With Devo, you can collect infrastructure metrics from on-prem and cloud environments and combine them with logs for troubleshooting. Devo also measures user response time and supports distributed tracing for APM use cases. Devo includes the Security Operations application for SIEM use cases including detections, threat hunt capabilities, entity analytics, case management, and machine learning models to spot anomalous behavior. The platform also includes the Service Operations application, which provides dependency and topology mapping, reporting, machine learning-based analytics, and digital experience monitoring.

Devo does not include an integrated SOAR platform as part of its solution but does integrate with solutions such as Palo Alto's XSOAR product.

Features	**Elastic**	**Splunk**	**Devo**
Centralized logging	X	X	X
Real-time search			X
Native multitenancy			X
400 days hot storage (included)			X
Ingest & search parellelization			X
Infrastructure metrics	X	X	X
Network flows	X	X	X
Distributed tracing	X		X
Custom dashboards	X	X	X
Multicloud	X	X	X
Entity behavior analytics	X	X	X
ML-based anomaly detection	X	X	X
Custom ML model support	X		X
Threat hunt	X	X	X
Threat detection	X	X	X
Case management	X	X	X
RESTful API	X	X	X
Incident timelines	X	X	X
Endpoint security (included)	X		
Totals	**15**	**12**	**18**

#3
Comparing the Costs of Elastic, Splunk, and Devo

It is challenging to compare the costs of these solutions because each company prices its products using different methods. Therefore, rather than compare specific costs, we will compare the pricing models themselves for fairness, simplicity, and predictability.

ELASTIC COST MODEL

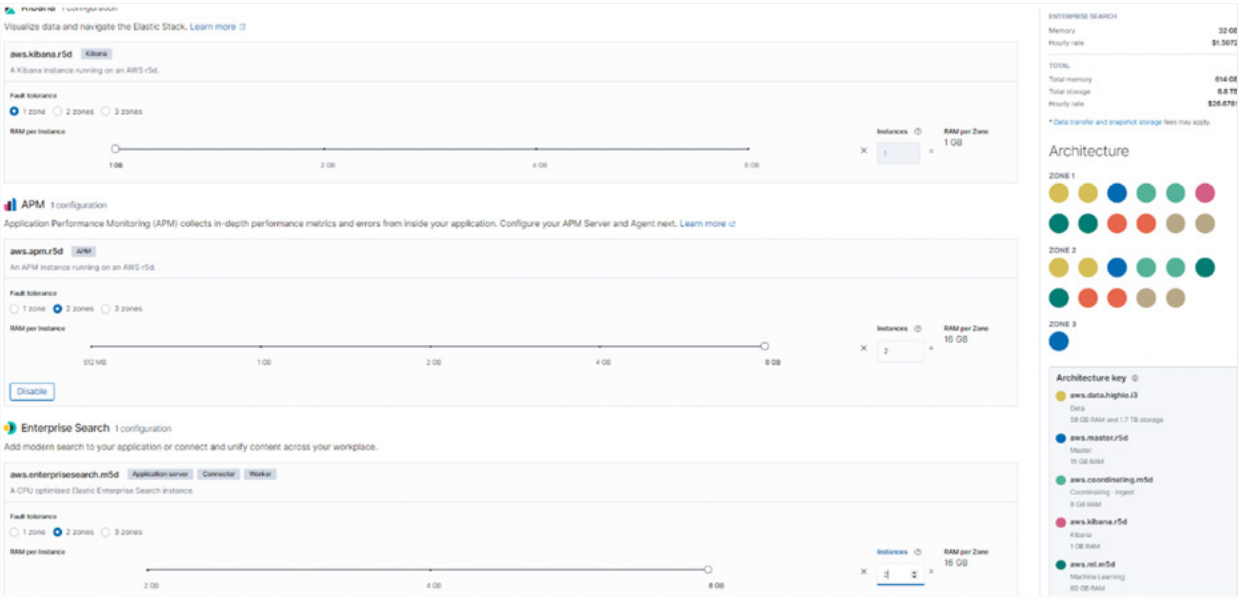
Elastic's cost model is based on infrastructure costs for compute, storage, and memory plus three tiers of support. Elastic's underlying architecture makes it challenging to size the solution for your environment. There is a cost estimator tool on Elastic's website: **https://cloud.elastic.co/pricing**.

There are different estimator tools for "Observability," "Security," and "Classic ELK Stack." You have the option to pick your cloud provider (AWS, Azure, GCP), and availability zone. You then need to decide how many CPUs, how much memory, and how many instances you need. Next, you must select if you require fault tolerance in multiple zones. This process must be repeated for additional components such as APM. Here is a screenshot of Elastic's price estimator tool.

As mentioned above in the Architecture section, Elastic requires a significant number of infrastructure components. Deploying it across multiple availability zones for fault tolerance doubles this already large infrastructure count. Lastly, because Elastic's indexes don't compress well, you also need a lot of storage—even when storing just a short time period of data. The final component is Elastic's levels of support, which can be found here: **https://www.elastic.co/pricing/**. Note that you need their highest support package to get access to their Endpoint Security capabilities. So, even though Elastic is open source, the cloud infrastructure costs can add up quickly.

SPLUNK COST MODEL

Splunk's pricing models are complicated and depend on many options. Broadly speaking, they break down into ingest pricing or resource pricing. The resource based pricing model is based mostly on the compute power for your searches. Compute resources are based on a combination of your total logical cores, multiplied by "a premium data %," and added to your premium cores. You also need to be very aware of what searches you regularly run and when those searches run, since you'll need to allocate dedicated compute power for those priority searches. You also have to go through this exercise for Splunk's core product, IT Service Intelligence, and Enterprise Security individually. The idea is that compute power is less important for ingestion compared to searches—so paying for compute cores dedicated to searching is a better value. But again, it's hard to know what data is important until you need it, and this model could result in slower performance for data that you hadn't already designated as "premium." Lastly, if you don't add compute power as your total amount of data volume increases, it most likely will result in slower search times as data volume grows. This resource based pricing model is new for Splunk and it remains to be seen how popular it will be.



Kibana 1 configuration

Visualize data and navigate the Elastic Stack. Learn more

aws.kibana.r5d Kibana

A Kibana instance running on an AWS r5d.

Fault tolerance

● 1 zone ○ 2 zones ○ 3 zones

RAM per instance: 1 GB

Instances: 1 × = RAM per Zone 1 GB

APM 1 configuration

Application Performance Monitoring (APM) collects in-depth performance metrics and errors from inside your application. Configure your APM Server and Agent next. Learn more

aws.apm.r5d APM

An APM instance running on an AWS r5d.

Fault tolerance

○ 1 zone ● 2 zones ○ 3 zones

RAM per instance: 8 GB

Instances: 2 × = RAM per Zone 16 GB

Disable

Enterprise Search 1 configuration

Add modern search to your application or connect and unify content across your workplace.

aws.enterprisesearch.m5d Application server Connector Worker

A CPU optimized Elastic Enterprise Search instance.

Fault tolerance

○ 1 zone ● 2 zones ○ 3 zones

RAM per instance: 8 GB

Instances: 2 × = RAM per Zone 16 GB

ENTERPRISE SEARCH

Memory	32 GB
Hourly rate	$1.5072

TOTAL

Total memory	614 GB
Total storage	6.8 TB
Hourly rate	$26.8781

* Data transfer and snapshot storage fees may apply.

Architecture

Architecture key

- ● aws.data.highio.i3 — Data — 58 GB RAM and 1.7 TB storage
- ● aws.master.r5d — Master — 15 GB RAM
- ● aws.coordinating.m5d — Coordinating · Ingest — 8 GB RAM
- ● aws.kibana.r5d — Kibana — 1 GB RAM
- ● aws.ml.m5d — Machine Learning — 60 GB RAM

Splunk's pricing models are complicated and depend on many options.

Historically, Splunk's pricing model has been based on data ingest volume. This is the pricing model most customers currently have. But this model also has many factors that come into play, and each one adds to your total cost. Splunk charges extra for each "premium application," such as IT Service Intelligence and Enterprise Security, and the cost of the premium application rises as the volume of data increases. Splunk also charges extra to encrypt data at rest, and the more data you have the higher the expense. They charge extra for additional storage—in 500GB blocks—to store data for longer historical periods. And all of these individual charges have two tiers based on whether you want the "Standard" or "Premium" plan. Before jumping to Splunk Cloud be sure to fully understand which applications you will need to address your use cases, how much historical data you require to be stored hot, and account for the cost of encrypting your data at rest, as well as support costs.

DEVO COST MODEL

Devo pricing is the simplest to understand and predict. Devo charges based on data ingestion per day averaged over a month-long period. That price includes all functionality, as well as encrypting data at rest. You receive access to the Security Operations application for SecOps use cases, the Service Operations application for ITOps use cases, and centralized log management for all other needs. There are no extra charges for adding users or dashboards. You receive 400 days of hot searchable storage included in the price. The only add-on option for Devo is to replicate data across availability zones. This pricing model makes Devo the most cost-effective of the three vendors.

PRICING MODEL RANKINGS

Splunk's approach to charging for almost every single feature makes it the least attractive option for potential buyers. Splunk's numerous additional charges for features such as encrypting data at rest can rapidly inflate the total cost and make it the most expensive solution on the market.

Elastic's model is complicated because it is so difficult to estimate the infrastructure necessary to support your environment and predict how those needs will grow over time. A word of caution with Elastic is that it needs a significant amount of cloud infrastructure, and the company passes these costs on to customers. Retaining more than 30 days of hot data on Elastic also consumes an expensive amount of storage.

Devo delivers all of the functionality provided by the other two vendors, but with a more efficient architecture and always-hot data access. This is due to the unique architecture detailed above. The Devo pricing model also makes it easy to predict and budget. The Devo UI clearly shows how much data you are ingesting per day, and it is easy to spot growth trends that enable you to predict month-to-month ingest rates. Devo customers never need to worry about adding virtual cores or storage or RAM—these are provided by Devo as part of the SaaS. Buyers never need to worry about under-sizing or over-sizing their cloud infrastructure. Devo handles all aspects of managing the environment, so you can focus on leveraging your data.

Devo delivers all of the functionality provided by the other two vendors, but with a more efficient architecture and always-hot data access.

CONCLUSIONS

Centralized log management has proven its value in a variety of use cases from ITOps to SecOps and more. Although centralized log management Solutions have always posed challenges, today's SaaS delivery model removes most of them and makes it easier than ever to deploy and run a centralized log management solution. For most medium-sized to large buyers with multiple environments (datacenters and multiple-cloud environments), Devo is the best choice for centralized log management because it delivers the most modern and efficient architecture, offers a rich feature set, and has the most attractive cost model.

Devo
255 Main Street
Suite 702
Cambridge, MA 02142

Devo is the only cloud-native logging and security analytics platform that releases the full potential of your data to empower bold, confident action. With unrivaled scale to collect all of your data without compromise, speed to give you immediate access and answers, and clarity to focus on the signals that matter most, Devo is your ally in protecting your organization today and tomorrow. Devo is headquartered in Cambridge, Mass. Learn more at **www.devo.com**.