Stopping Cybersecurity Threats: Why Databases Matter

scylladb.com/2021/11/11/stopping-cybersecurity-threats-why-databases-matter/

November 11, 2021

From intrusion detection, to threat analysis, to endpoint security, the effectiveness of cybersecurity efforts often boils down to how much data can be processed — in real time — with the most advanced algorithms and models.

Many factors are obviously involved in stopping cybersecurity threats effectively. However, the databases responsible for processing the billions or trillions of events per day (from millions of endpoints) play a particularly crucial role. High throughput and low latency directly correlate with better insights as well as more threats discovered and mitigated in near real time. But cybersecurity data-intensive systems are incredibly complex; many span 4+ data centers with database clusters exceeding 1000 nodes and petabytes of heterogeneous data under active management.

How do expert engineers and architects at leading cybersecurity companies design, manage, and evolve data architectures that are up to the task? Here's a look at 3 specific examples.

Accelerating real-time threat analysis by 1000% at FireEye

Cybersecurity use case

FireEye's Threat Intelligence application centralizes, organizes, and processes threat intelligence data to support analysts. It does so by grouping threats using analytical correlation, and by processing and recording vast quantities of data, including DNS data, RSS feeds, domain names, and URLs. Using this array of billions of properties, FireEye's cybersecurity analysts can explore trillions of questions to provide unparalleled visibility into the threats that matter most.

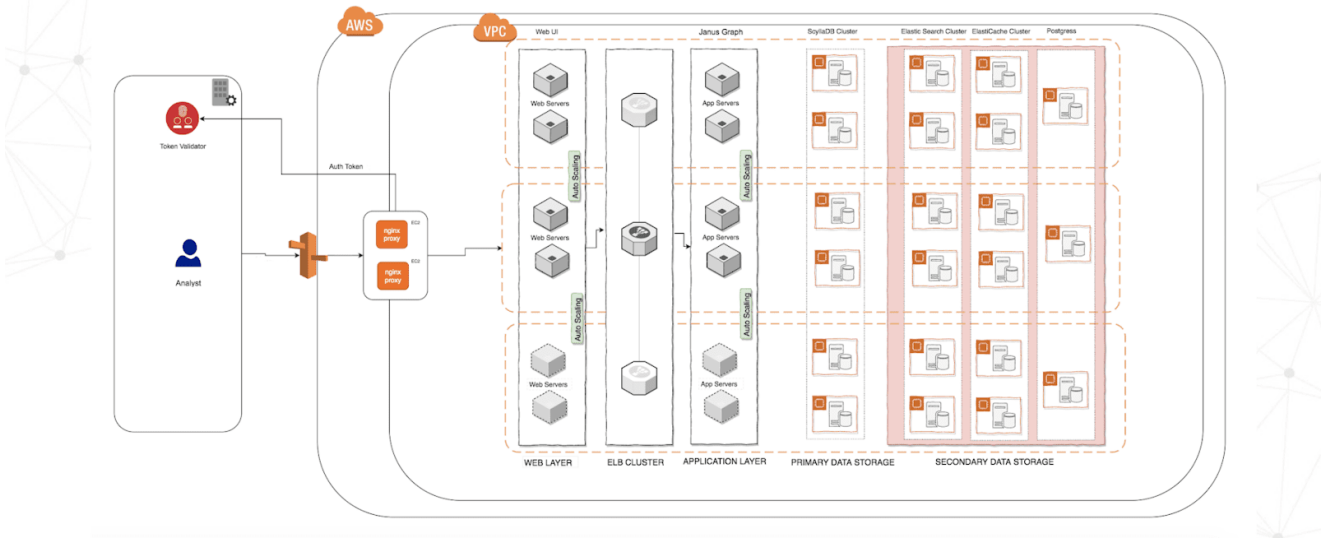
Database challenge

Their legacy system used PostgreSQL with a custom graph database system to store and facilitate the analysis of threat intelligence data. As the team of analysts grew into the hundreds, system limitations emerged. The graph size grew to 500M nodes, with around 1.5B edges connecting them. Each node had more than 100 associated properties accumulated over several years. The PostgreSQL-based system became slow, proved difficult to scale, and was not distributed or highly available. FireEye needed to re-architect their system on a new technology base to serve the growing number of analysts and the businesses that rely on them.

Database strategy

To start, the team evaluated several graph databases and selected JanusGraph. FireEye's functional criteria included traversing speed, full/free text search, and concurrent user support. Non-functional criteria included requirements for high availability and disaster recovery, plus a pluggable storage backend for flexibility. The team felt that JanusGraph met these criteria well, and also appreciated its user-controllable indexing, schema management, triggers, and OLAP capabilities for distributed graph processing.

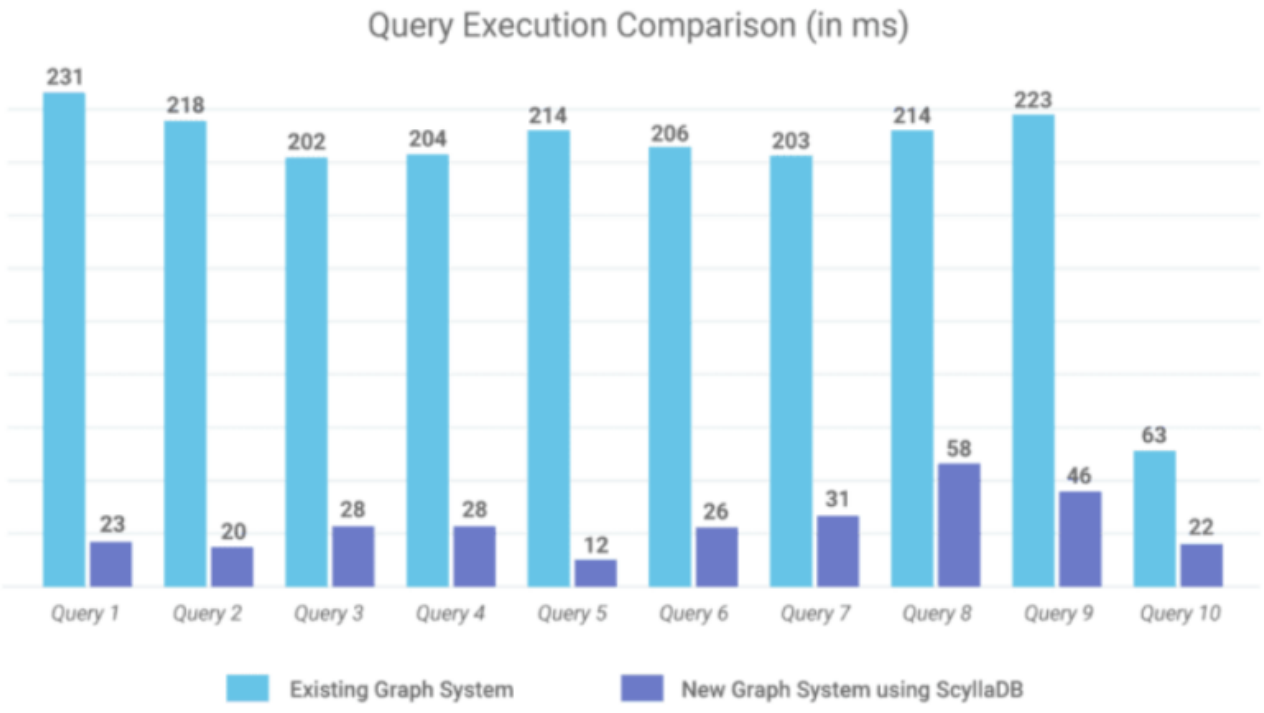
Next, they shifted focus to evaluating compatible backend storage solutions. With analysis speed top of mind, they looked past Java-based options. FireEye selected ScyllaDB based on its raw performance and manageability. The FireEye team chose to run ScyllaDB themselves within a secure enclave guarded by an NGINX gateway. Today, the ScyllaDB solution is deployed on AWS i3.8xlarge instances in 7-node clusters. Each node is provisioned with 32 CPUs, 244MB of memory, and 16TB SSD storage.



System architecture diagram – provided by FireEye

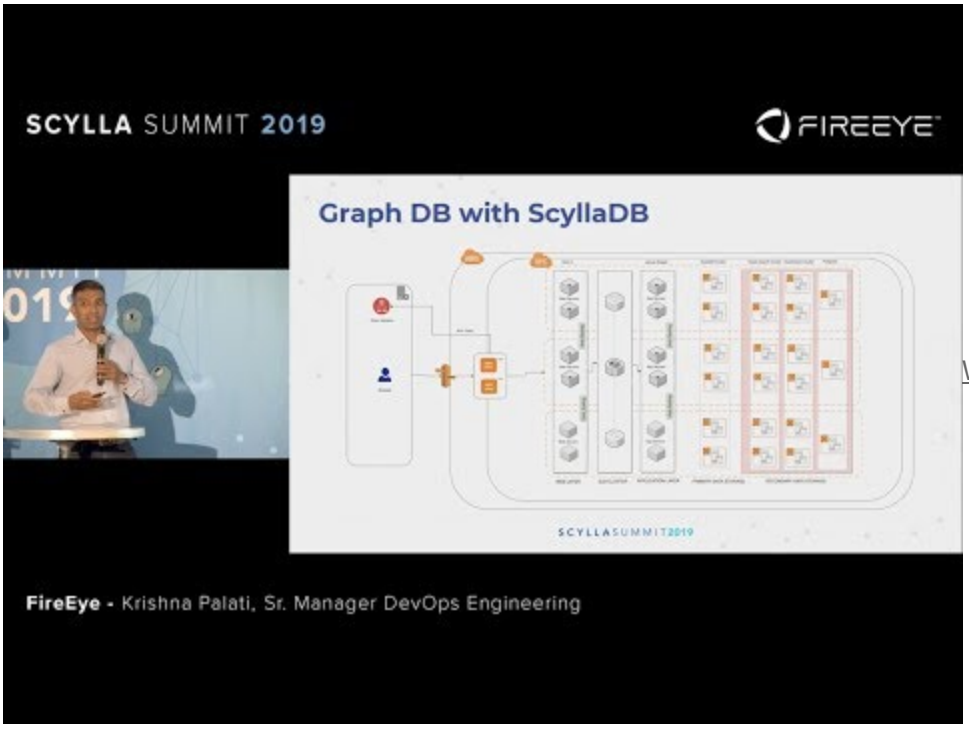
Impact

With the new JanusGraph + ScyllaDB system, FireEye achieved a performance improvement of 100X. For example, a query traversing 15,000 graph nodes now returns results in 300ms (vs 30 seconds to 3 minutes).



Query execution comparison diagram – provided by FireEye

Moreover, they were able to dramatically slash the storage footprint while preserving the 1000-2000% performance increase they had experienced by switching to ScyllaDB. Ultimately, they reduced AWS spend to 10% of the original cost.



Watch Video At:

https://youtu.be/3CuOXfuORAg

Read more about the role of databases in FireEye's cybersecurity efforts

Scaling security from 1.5M to 100M devices at Lookout

Cybersecurity use case

Lookout leverages artificial intelligence to provide visibility and protection from network threats, web-based threats, vulnerabilities, and other risks. To protect an enterprise's mobile devices, they ingest device telemetry and feed it across a network of services to identify risks. Low-latency is key: if a user installs an app with malware and it takes minutes to detect, that device's information is already compromised.

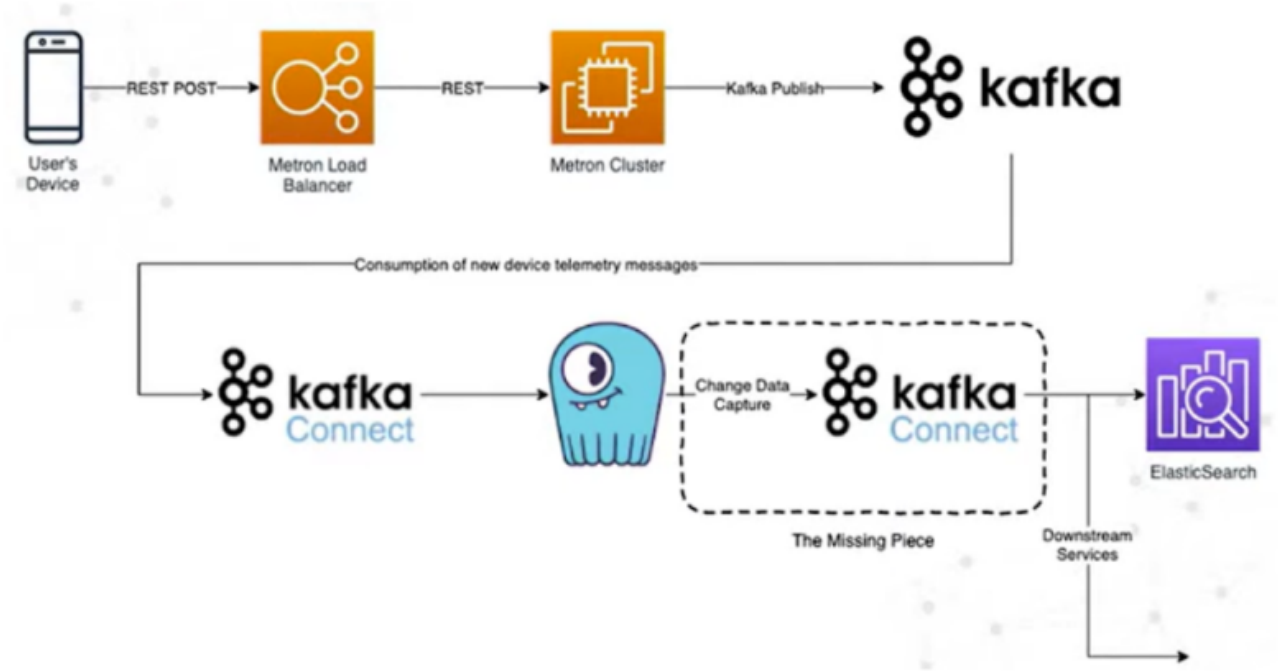
Database challenge

Lookout needed to scale from supporting 1.5 million devices to 100 million devices without substantially increasing costs. This required ingesting more and more telemetry as it came in. They needed a highly scalable and fault-tolerant streaming framework that could process device telemetry messages and persist these messages into a scalable, fault-tolerant persistent store with support for operational queries.

Their existing solution involved Spark, DynamoDB, and ElasticSearch. However, DynamoDB alone would cost them about $1M per month if they scaled it for 100M devices. Also, DynamoDB lacked the required level of sorting for time series data. With DynamoDB, they had to use a query API that introduced a significant performance hit, impacting latency.

Database strategy

Lookout replaced Spark with Kafka and migrated from DynamoDB to ScyllaDB. Kafka Connect servers send data from Kafka into ScyllaDB, then it's pushed into ElasticSearch.

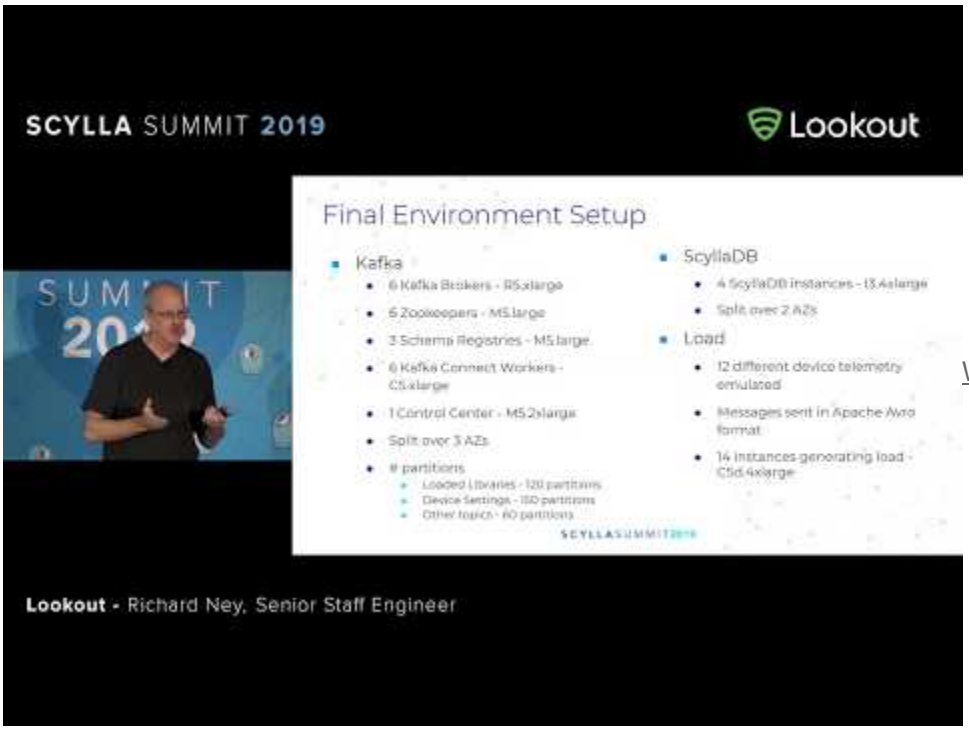


System architecture diagram – provided by Lookout

They also discovered that Kafka's default partitioner became less and less efficient with sharding as the number of partitions grew, so they replaced the default Kafka sharding function with a murmur3 hash and put it through a consistent hashing algorithm (jump hash) to get an even distribution across all partitions.

Impact

Lookout met its goal of growing from 1.5 million devices to 100 million devices while reining in costs. What would have cost them nearly $1M per month with DynamoDB cost them under $50K per month with ScyllaDB. They also achieved the low latency vital to effective threat detection. Their message latency is currently in the milliseconds, on average.



Watch Video At:

https://youtu.be/VHIIjLMAT5s

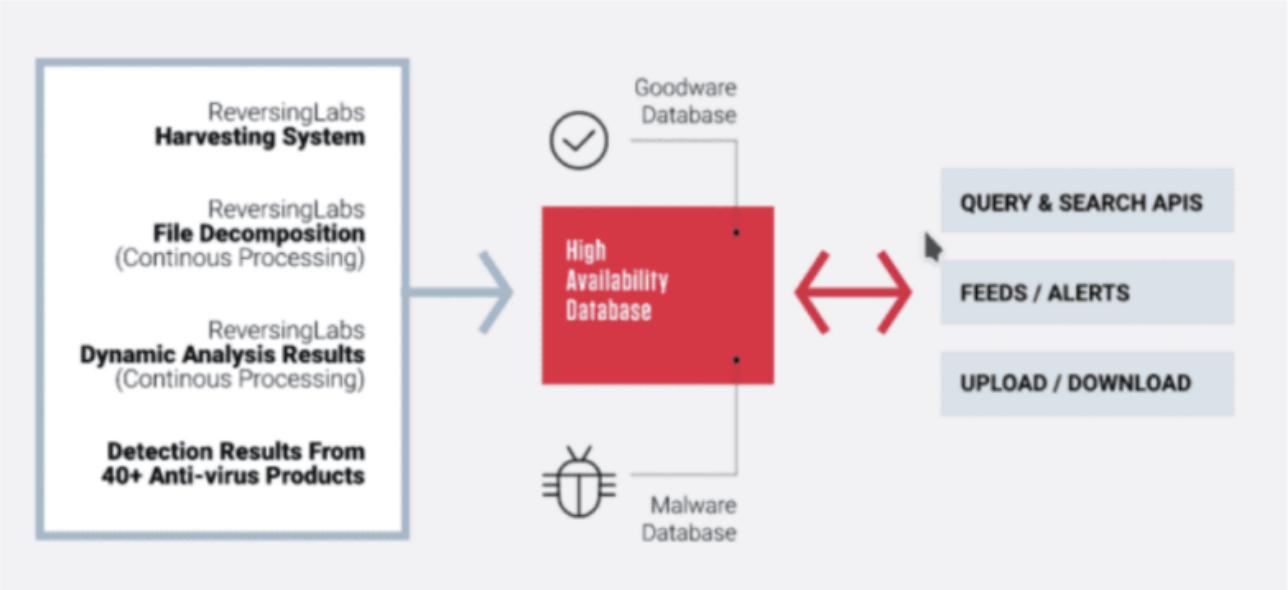
Achieving database independence at ReversingLabs

Cybersecurity use case

ReversingLabs is a complete advanced malware analysis platform that speeds destructive file detection through automated static analysis — enabling analysts to prioritize the highest risks with actionable detail in milliseconds. Their TitaniumCloud Reputation Services provides powerful threat intelligence solutions with up-to-date threat classification and rich context on over 20B goodware and malware files.

Database challenge

ReversingLabs' database stores the outcome of multiple analyses on billions of files so that end users can perform malware detection "in the wild": immediately checking if a file they encounter is known to be safe or suspected to be malware. If the file's status is unknown, it needs to be analyzed then entered into the ReversingLabs database. 50+ APIs and specialized feeds connect end users to that database.



System architecture diagram – provided by ReversingLabs

Years ago, ReversingLabs built their own highly available database for this application. It's a key-value store, based on the LSM Tree architecture. The database needed to insert/update and read large amounts of data with low latency and stable response times, and it achieved those goals.

However, as their business expanded, they wanted "database independence" to eliminate dependencies on the aging legacy database. They built a library that enabled them to connect to other databases without modifying their application layer. But connecting OSS or COTS solutions to meet their specialized needs wasn't exactly a plug-and-play endeavor. For example, they needed:

- Key-value native protobuf format
- LZ4 compression for reduced storage size
- Latency <2 ms
- Support for record sizes tanging from 1K to 500M

Database strategy

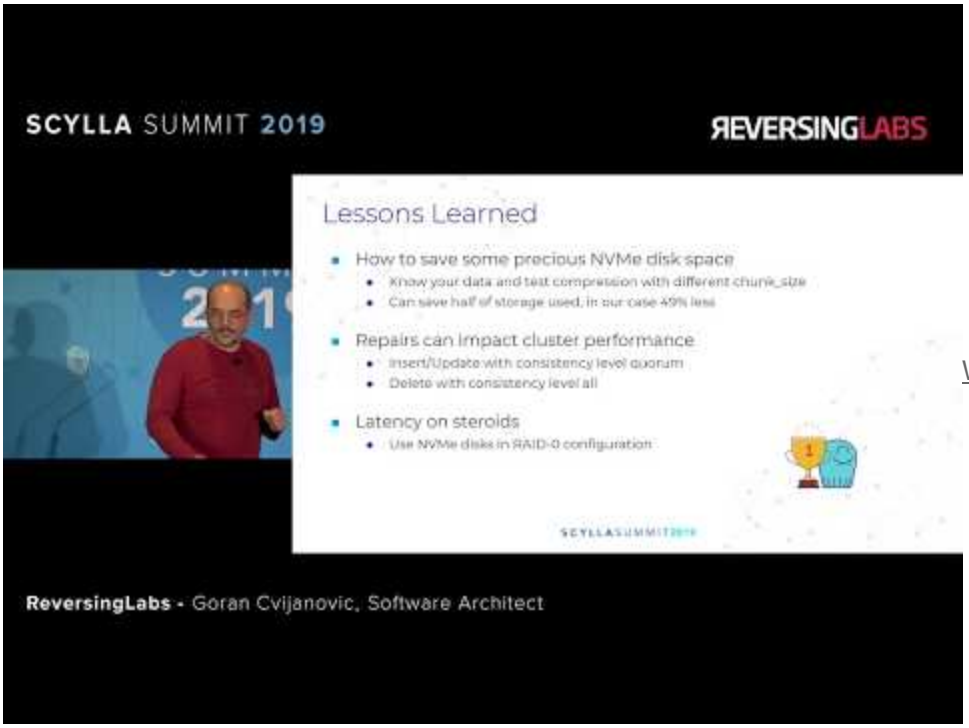
The first test of their database independence strategy was adopting ScyllaDB, with an 8 node cluster, for their 2 large volumes APIs. They were able to meet their specialized needs by applying strategies like:

- Using blobs for keys as well as values
- Tuning the chunk size to improve compression 49%
- Improving performance by using NVMe disks in RAID 0 configuration and doing inserts/updates and reads at consistency level quorum

Impact

ReversingLabs successfully moved beyond their legacy database, with a highly available and scalable system that meets their specialized needs. Within the ScyllaDB database, file reputation queries had average writes latencies of <6 milliseconds and sub-millisecond average reads. For p99 (long-tail) latencies, they achieved <12 millisecond writes, and <7 millisecond reads.

In end-to-end workflow testing (including and beyond the database), average latencies were less than 120 milliseconds, and p99 latencies were 166 milliseconds. This included the user request, authentication process, request validation, the round-trip time to query the database, to format and then finally send the response to the application, scaled to 32 workers, all in parallel.



Watch Video At:

https://youtu.be/TJyThkI3-0s

Read more about the role of databases in ReversingLabs' cybersecurity efforts

Databases: what's essential for cybersecurity

To wrap up, here's a quick recap of the core database capabilities that have helped these and many other companies stop cybersecurity threats faster and more accurately:

- Consistent low-latency performance for real-time streaming analytics — to identify and prevent digital threats in real-time
- Extreme throughput and rapid scaling — to ingest billions of events across threat analysis, malware protection, and intrusion detection

- Lower total cost of ownership, reduced complexity, automated tuning, and DevOps-friendly operational capabilities — to free resources for strategic cybersecurity projects
- The ability to run at extremely high levels of utilization — to ensure consistent performance and ultra-low latency without overprovisioning and waste
- Cloud, on-premises, and a variety of hybrid topologies to support data distribution and geographic replication — for high availability, resilience, and instant access to critical data anywhere in the world

Want advice on whether a database like ScyllaDB is a good fit for your environment and your use case? Chat with us or sign up for a free technical 1:1 consultation with one of our solution architects.

Schedule a technical consultation

JanusGraphelasticsearchDynamoDBApache Sparkcybersecurityintrusion detectionPostgreSQLFireEyeLookoutReversingLabsApache Kafkathreat intelligencethreat analytics