White Paper

Information Security | Machine Learning

October 2020

IT@Intel: Building a Modern, Scalable Cyber Intelligence Platform with Apache Kafka

Our Apache Kafka data pipeline based on Confluent Platform ingests tens of terabytes per day, providing in-stream processing for faster security threat detection and response

Intel IT Authors

Ryan Clark
Information Security Engineer

Jen Edmondson
Product Owner

Dennis Kwong
Information Security Engineer

Jac Noel
Security Solutions Architect

Elaine Rainbolt
Industry Engagement Manager

Paul Salessi
Information Security Engineer

Intel IT Contributors

Victor Colvard
Information Security Engineer

Juan Fernandez
Technical Solutions Specialist

Frank Ober
SSD Principal Engineer

Table of Contents

Executive Summary

Advanced cyber threats continue to increase in frequency and sophistication, threatening computing environments and impacting businesses' ability to grow. More than ever, large enterprises must invest in effective information security, using technologies that improve detection and response times. At Intel, we are transforming from our legacy cybersecurity systems to a modern, scalable Cyber Intelligence Platform (CIP) based on Kafka and Splunk. In our 2019 paper, Transforming Intel's Security Posture with Innovations in Data Intelligence, we discussed the data lake, monitoring, and security capabilities of Splunk. This paper describes the essential role Apache Kafka plays in our CIP and its key benefits, as shown here:

Apache Kafka is the foundation of our CIP architecture. We achieve economies of scale as we acquire data once and consume it many times. Simplified connection of data sources helps reduce our technical debt, while filtering data helps reduce costs to downstream systems.

Intel vice president and Chief Information Security Officer, Brent Conran, explains, "Kafka helps us produce contextually rich data for both IT and our business units. Kafka also enables us to deploy more advanced techniques in-stream, such as machine-learning models that analyze data and produce new insights. This helps us reduce mean time to detect and respond; it also helps decrease the need for human touch. Kafka technology, combined with Confluent's enterprise features and high-performance Intel architecture, support our mission to make it safe for Intel to go fast."

Data Silos Impede Response

At Intel, our Information Security organization keeps the enterprise secure, while managing legal compliance worldwide. It requires a large number of people, applications, databases, and analytics capabilities. In the past, we were not always well integrated, which led to silos in multiple locations focused on specific information. Intel IT is not alone. When left unaddressed, these disparate silos can lead to poor or no data integration, dead-ends, inconsistencies, inaccurate interpretations, and significant technical debt. In these scenarios, organizations might spend more time maintaining outdated legacy systems than preparing for emerging threats. Siloed and disparate systems cannot keep pace with the rapidly changing threat landscape. They hinder efficient prevention, detection, and response to threats and vulnerabilities. We needed the ability to connect and integrate on a single platform that enables us to use our data to its greatest advantage.

Defining a Modern, Scalable Architecture

Our Information Security organization works in concert to operate on data, construct detection logic, and create dashboards to quickly identify, triage, and mitigate threats. Effectively communicating event data in a timely manner is critical. Our solution requires a resilient, highly available, scalable platform capable of ingesting multiple TBs of data per day from many sources, and in near real time. Our Cyber Intelligence Platform (CIP) goes beyond traditional security information and event management capabilities to continually increase data effectiveness across the organization. That includes vulnerability management, security compliance and enforcement, threat hunting, incident response, risk management, and other activities. Through modern, industry-leading technologies, our new CIP increases the value of our data by making it more usable and accessible to everyone in the organization (see Figure 1).

> "Kafka also enables us to deploy more advanced techniques in-stream, such as machine-learning models that analyze data and produce new insights."
>
> *–Brent Conran, Vice President and CISO, Intel*

Selecting an Apache Kafka Supplier

Before choosing Confluent as our technology partner, we evaluated several publish and subscribe (Pub/Sub) data pipeline technologies for our CIP solution. We needed the capability to ingest massive amounts of data, produce feeds, and perform in-stream data transformations (also known as stream processing) with high throughput and low latency. Essentially, we needed a circulatory system that could filter, refine, and improve data before it is consumed. We saw that the ability to acquire data once and consume it many times would deliver economies of scale to Information Security, IT operations, and Intel's business units. We decided Apache Kafka was the best technology for these desired capabilities.

Information Security's mantra is to buy before build whenever possible, and we knew we needed a Kafka supplier to help with the typical pain points and risks that accompany pure open source software. Upon scanning the market, it quickly became obvious that Confluent was the leader in supplying value-add platform capabilities, in addition to enterprise software support for Kafka. Other options claimed to support Kafka; however, because they were also focused on a broader set of software, such as Hadoop, Spark, and Apache NiFi, their expertise with Kafka was diluted. Conversely, Confluent was laser-focused on Kafka software and its surrounding community, making them our partner of choice.

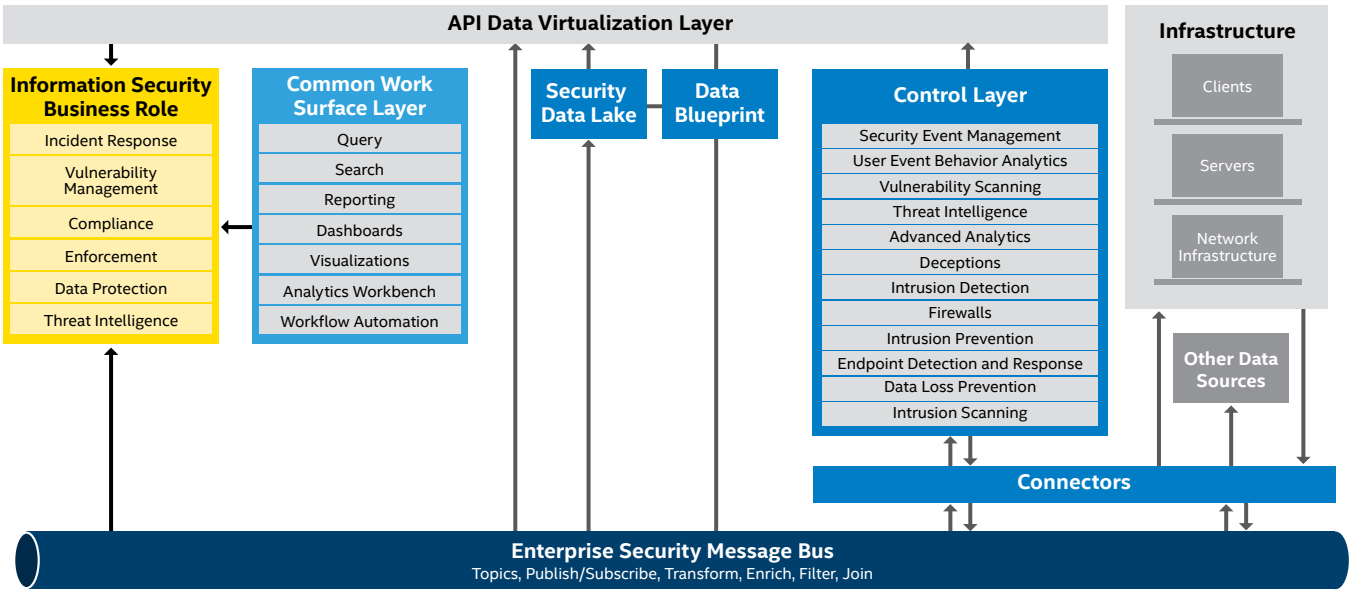


Figure 1. Our new CIP requires modern technologies that quickly and easily consume new data sources, employ machine learning, and extend capabilities.

The combination of open source Apache Kafka with Confluent as our technology partner enabled Information Security to quickly realize our objectives and vision for the Enterprise Security Message Bus in our CIP solution architecture (see Figure 2).

Confluent's Enterprise Capabilities

Confluent provides enterprise capabilities beyond open source Apache Kafka that make it more powerful, manageable, maintainable, and easy to use. Some of these key capabilities are:

- Confluent Control Center for monitoring our Kafka clusters
- Replicator and Multi-Region Clusters for multiple data center design and fault tolerance
- Auto Data Balancer for cluster rebalancing and easy scale out, when needed
- Enterprise-grade security including Role-based Access Control and audit logs
- 100+ pre-built connectors to common sources and sinks
- Technical Account Manager for feature prioritization and guidance
- Architecture and design partnership via a Confluent Resident Architect and "health check" engagements
- Ongoing training available in real time (for example, webinars) and on demand
- Availability of 24x7x365 support
- Partnership for collaborating on new capabilities, such as security features

Benefits of Apache Kafka

Apache Kafka and Confluent Platform offer several benefits:

- **Economies of scale.** The concept of acquiring data once and consuming it many times is extremely powerful in describing the benefits of Kafka. When data is produced to Kafka once, the options for consuming that data become endless. Gone are the days of many point-to-point and custom connectors to integrate disparate technologies (see Figure 3). It is now more common to seamlessly share data across multiple people, applications, and systems. It also makes it easier to monitor, build, and maintain data pipelines, all in one place.
- **Operate on data in-stream.** Rapid threat detection and response is a primary goal for the Information Security team, whose successes are often measured by indicators like mean-time-to-detect and mean-time-to-respond. Intel is no different. Now, with the ability to operate on data in-stream, we can identify threats in near real time.
- **Reduce downstream costs.** The ability to filter data in-stream can help reduce costs. For example, if the first operation you plan to perform is filtering duplicate data, it may be better to do it in-stream, thus reducing compute and ingestion costs in downstream applications like Splunk.
- **Reduce technical debt.** We eliminated hundreds of legacy and custom point-to-point connectors for our security capabilities and analytic solutions. This has significantly reduced the typical care and maintenance costs related to supporting these disparate technologies (see Figure 3).
- **Generate contextually rich data.** Transforming data in-stream enables us to deliver contextually rich, clean, and high-value data to our Information Security teams, business units, and functional organizations. For example, by performing streaming joins and enrichments, we can integrate multiple, disparate, security data sources in near real time. This transformed data enhances downstream applications' abilities to make faster and/or more accurate decisions.

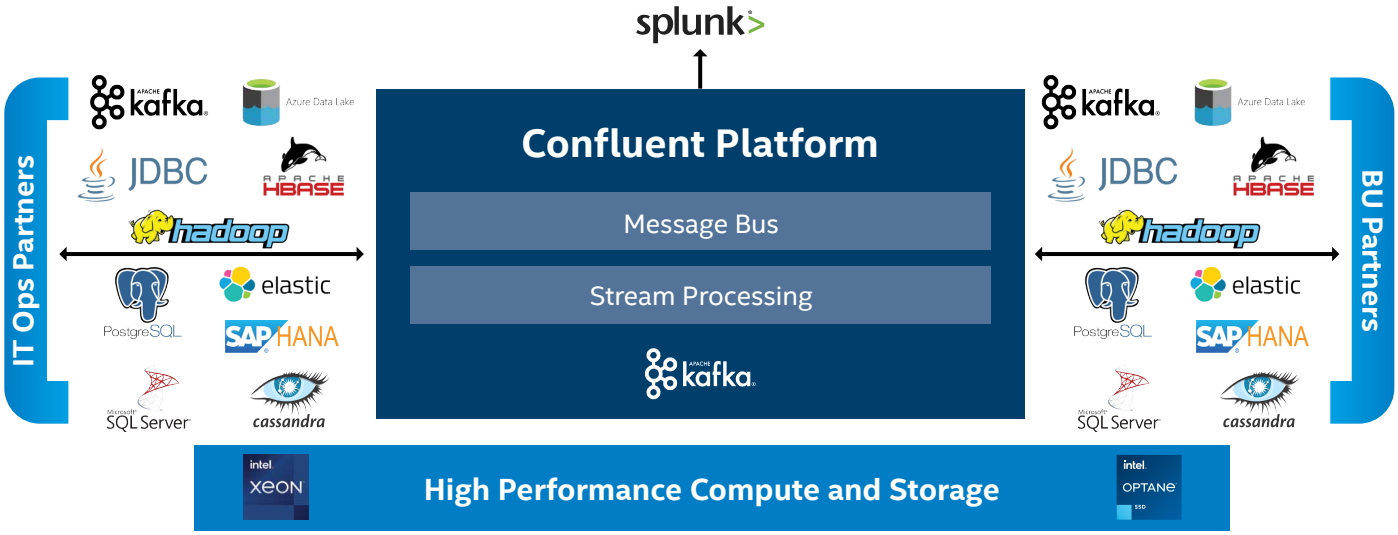


Figure 2. Confluent Platform supports our vision to generate contextualized data once and consume it many times, creating economies of scale across many Intel® organizations.

- **Global scale and reach.** Kafka's distributed bus technology connects data centers across the globe, in the cloud, in multiple regions and countries, and all the way to the edge. Message bus and queuing technologies like to integrate with each other. You can easily integrate Kafka with traditional message queue (MQ) technologies or more modern technologies like Azure Event Hubs, AWS Kinesis, GCP Pub/Sub, Confluent Cloud, and more.
- **Always on.** Kafka's solution architecture enables the Confluent Platform to be an always-on service, eliminating the days of managed downtime. Notifications to all producers and consumers of an impending reboot or upgrade are no longer necessary. The ability to do live rolling upgrades and restarts allows you to perform the actions seamlessly. In addition, producers and consumers can go on and offline (planned or unplanned downtime) without affecting each other or Kafka. Kafka continues to record, persist, and replicate data even when elements are offline. When elements come back online, they can independently return to where they left off.
- **Modern architecture with a thriving community.** The Apache Kafka community is very active and continuously delivering new and innovative features, APIs, abstraction layers, and ways to connect to other systems. Kafka features and use cases are constantly and rapidly emerging thanks to the attentive community, of which Confluent is an active leader.
- **Apache Kafka leadership through Confluent expertise.** The relationship between Intel and Confluent has enabled us to more rapidly meet our goals and deliver on the promise and vision of our reference architecture. Confluent is an industry leader in supplying solutions and support for Kafka. They have proven to be laser-focused on the solution stack and the community that surrounds it.

Data Transformation Using Apache Kafka

Historically, data is connected using point-to-point integration, which can add significant complexity and become difficult to maintain over time. Sustaining point-to-point integration also increases overall technical debt. Adding data sources is time-consuming, monitoring and governance is difficult, and orchestration is often unachievable. However, with the Kafka message bus, data is acquired once and consumed by multiple applications (see Figure 3).

The message bus creates a data abstraction layer for upstream producers and downstream consumers, and it requires less work to maintain. Customers no longer need to develop their own point-to-point integrations; they simply subscribe to the topics they need. Data sources and applications can be added or removed without disrupting the entire system.

Pub/Sub also enables publishers or "producers" to categorize data into classes for consumption by subscribers or "consumers." Through categorization, data topics are defined; in the Information Security environment, examples are context topics, detect topics, and prevent topics. With Kafka Streams, we can slice, filter, enrich, aggregate, and normalize data in-stream. This allows publishers and subscribers to work independently when editing and using information, ultimately increasing the overall value of the data. We can transform data by:

- **Enriching**. Enriching an IP address by associating a host name.
- **Joining**. Combining an event containing a user account with other data containing worker-specific information, such as a business unit. Or if there is a known threat with specific matches (such as a nefarious URL), we can branch the data into a new topic associated with traffic detail about that URL.
- **Slicing**. Selecting data based on details, such as date, site, or device.
- **Filtering**. Removing extraneous or redundant details, making it easier for consumers to use and lowering ingestion and data storage costs.
- **Parsing**. Evaluating a text string to test conformity to a structural pattern, such as validating the correct format for an email address.

Kafka Streams is an API/framework on top of Kafka that can be distributed across servers in its own cluster. Those servers include consumers that pull data from topics to perform the streams-processing tasks. The results use Kafka producers to write back to new topics or even downstream to other systems like databases and APIs.

Streams processing also allows us to modernize many of our data pipelines by replacing some traditional daily batch Extract-Transform-Loads (ETL) with close to real-time enrichment. We no longer need to define a schema for every data source, which gives us a more powerful tool for data engineering.

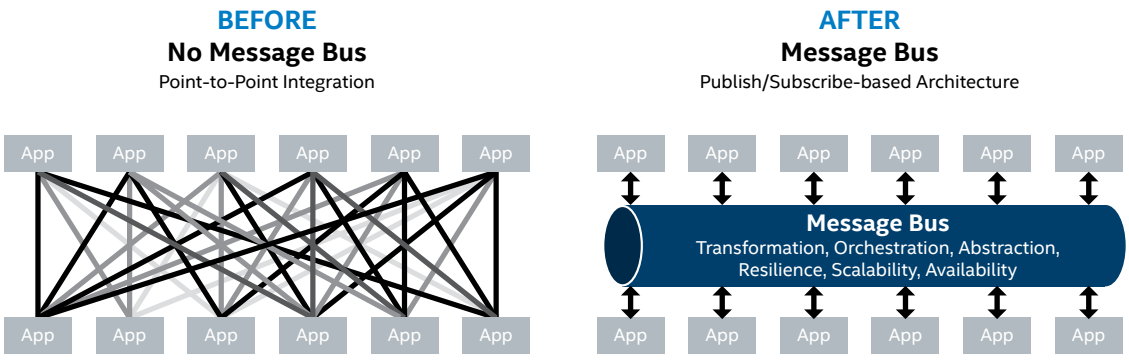


Figure 3. The message bus eliminates point-to-point integration, connecting applications across the enterprise to highly refined and relevant data.

Leveraging the power of Kafka Streams, we can do things like enrich a stream of vulnerability data by joining it with asset ownership data or filter the vulnerabilities by business unit based on IP subnets. This process of filtering and joining the data "in stream" results in new topics populated with contextually enriched data for a variety of systems, applications, and users to consume (see Figure 4).

Managing Our Kafka Cluster

Confluent Control Center (C3) monitors and administers Confluent Platform. It consumes topics to derive metrics, aggregate and perform calculations, and display the results through a web user interface. C3 leverages Kafka Streams in the backend to provide insights into the health of our Kafka platform, tracking some of our key metrics like consumer lag, throughput, and latency. With the ability to alert when metrics exceed certain thresholds, our mean-time-to-respond is reduced when handling anomalous events in the environment.

Securing Apache Kafka

Confluent Platform uses role-based and centralized access control and supports transport layer security (TLS) mutual authentication between brokers and clients with access control lists for topic-level security (see Figure 5). For example, a producer client may be assigned a single, specific topic to produce, yet not to consume.

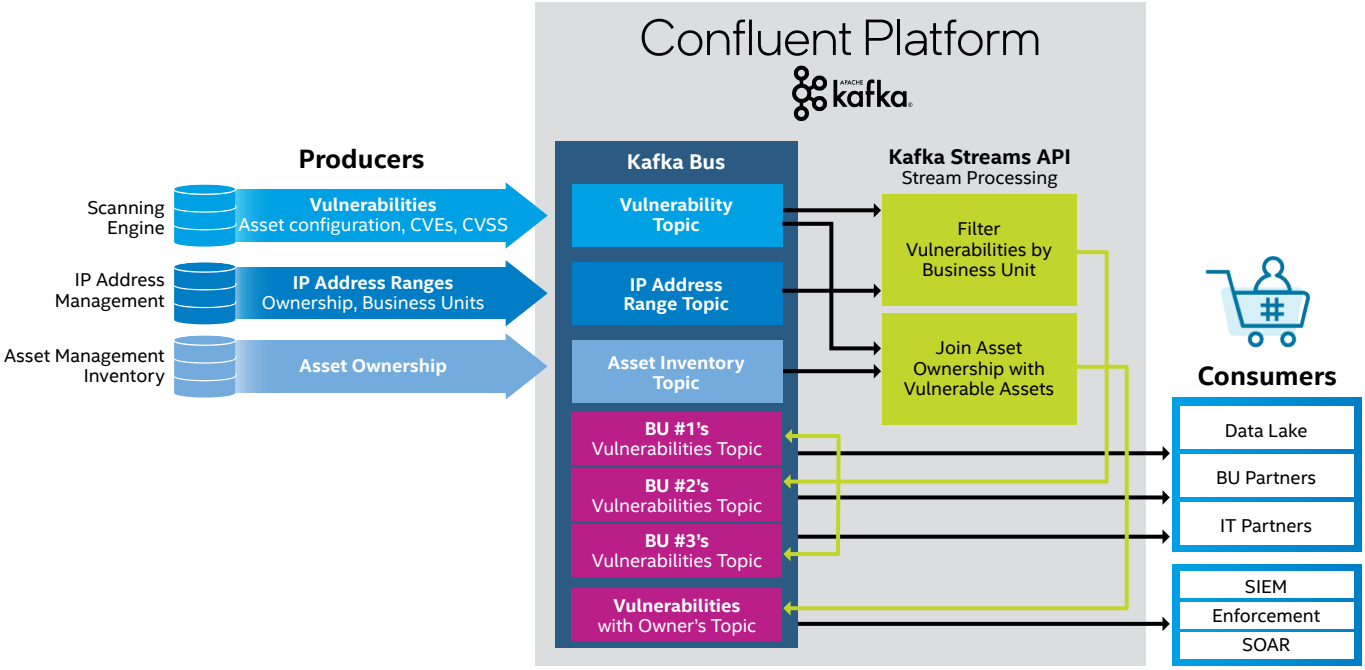


Figure 4. Our Confluent Platform contains hundreds of topics. In this example, we transform data in-stream and share with Information Security teams, IT operations, and business unit consumers.

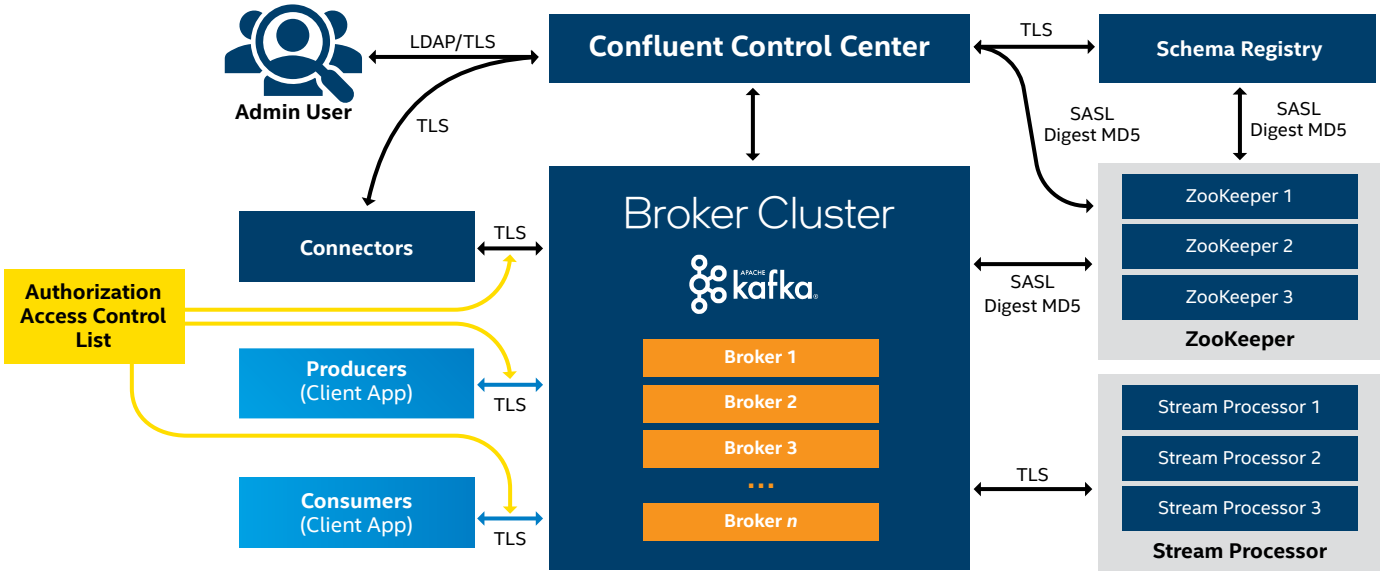


Figure 5. Confluent Platform uses role-based and centralized access control with topic-level security.

Improving Availability with Multi-Region Clusters

Identifying ways to improve data availability is one of the most important jobs of our Kafka architects and engineers. Rather than the traditional two-cluster approach to ensuring availability, we implemented Confluent's Multi-Region Clusters capability.

In the traditional model, data consumers use offsets to keep track of where in the stream they were located. In our previous architecture, the two Kafka cluster sites were isolated; while we replicated data across sites and maintained topic names, each cluster had its own offsets. For example, when a consumer pulled data from data center 1 Topic A and stopped at offset 100, attempts to fail-over to data center 2 Topic A/offset 100 were not always correct as offset ordering was not maintained.

Now, our Confluent Platform cluster spans two regions (or data centers), which eliminates the manual offset work and allows consumers to immediately return to where they left off. Because both sites are included in a single cluster, topics and offsets remain intact using asynchronous replication. Data center 2 may experience a slight latency, but it will correctly maintain offsets and make fail-over much easier (see Figure 6).

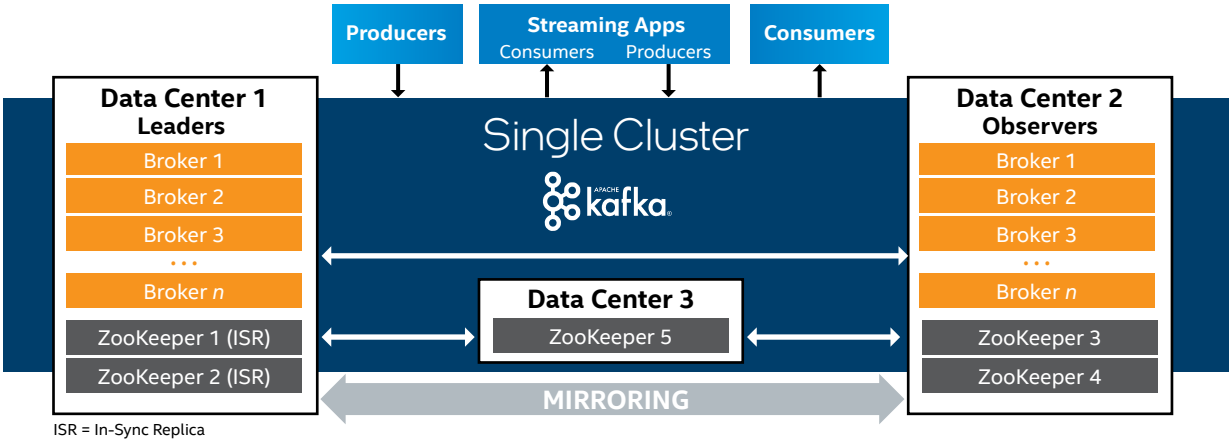


Figure 6. We use Confluent Platform to implement multi-region clusters for a more resilient architecture and significantly less manual work in fail-over events.

Intel and Confluent, Better Together

Information Security's data needs are similar to other business and IT organizations. Our data volume continues to increase and the velocity of data handling keeps accelerating. Our mission-critical applications need data shared between public and private services, database archives, SaaS applications, and thousands of device feeds. Their interdependencies make solutions difficult to maintain and create delays in business insights and automated response times. We also wanted 24x7x365 access to data in real time, without degrading its quality. This requires replicating and storing data in multiple locations. We also wanted to share the data from many sources and allow multiple applications to use it.

With Kafka, we can now easily add and remove producers, consumers, and topics. We're seeing system flexibility and cost reduction with optimized data stores. Like many other Kafka users, we're also slicing, filtering, enriching, aggregating, and normalizing data in-stream. Confluent's Control Center (C3) and Multi-Region Clusters help us manage our Kafka clusters for high availability and better performance. Our engineers and analysts can write rules-based logic to automate the mundane, such as filtering out false positives. Our data scientists and threat hunters are using tools like Kafka streams to hunt potential threats in real time, and our architecture team is adding new capabilities, like KSQL, to strengthen our position in the fight against cybercrime.

The benefits of Kafka and Confluent Platform to our CIP raised the visibility of Apache Kafka within Intel. The solution architects and engineers in Intel's Data Center Platforms Group are now working with Confluent to deliver higher performance, lower cost solutions. For example, Intel® Optane™ SSDs provide low latency, high throughput and 6.5X higher endurance than Intel® 3D NAND SSDs.[1,2] Another example is C3. Our testing revealed that 2nd Gen Intel® Xeon® Gold 6258R processors with 28 cores (56 virtual cores) provide the necessary performance for this demanding application.

Additional findings are available in the paper, "Enabling Real-Time Processing of Massive Data Streams." The paper is designed to help architects and engineers to:

- Make it easy to identify the infrastructure (compute, memory, network, and storage) required for today's Confluent Platform implementations.
- Provide sizing guidance to scale up or scale out Confluent Platform with business objectives that require increased endurance, low-latency, and high-throughput workloads.

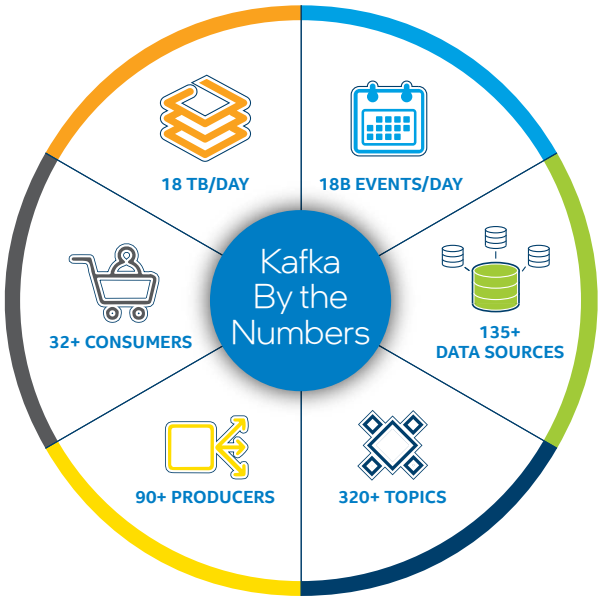
Pairing Confluent Platform's enterprise features and benefits with Intel® products and technologies provides organizations with a powerful solution that can generate contextually rich analytics from hundreds of data sources, across tens (even hundreds) of terabytes of data per day.

Sizing Apache Kafka Infrastructure

Intel's Information Security organization ingests approximately 18 TB of data per day—equivalent to 18 billion events—on the message bus. To handle this workload, we implemented a highly available multi-region Kafka cluster that includes 16 broker servers, 8 connect worker servers, 12 stream processor servers, 5 ZooKeepers, 4 schema registry servers, and 2 C3 servers. The cluster runs on a 10 GbE network across three sites (or data centers) with the third site hosting a ZooKeeper server as the ensemble tiebreaker. Apache Kafka is designed to take advantage of parallel processing and scale out with multiple servers and CPU cores.

The first Kafka brokers, connect workers, and stream processor servers we deployed were two-socket servers with 14-core Intel® Xeon® Gold 6132 processors. To expand compute and storage capacity, we added brokers and connect workers with two-socket servers built with 18-core Intel® Xeon® Gold 6140 processors. We also added stream processor servers with two-socket servers built with 24-core Intel® Xeon® Platinum 8160 processors and equipped them with 384 GB of memory and Intel® 3D NAND SSDs. To support the growth of additional Kafka Stream applications, we upgraded memory on the stream processor servers to 1 TB each.

In our experience, stream processor servers are memory-intensive because they run Kafka Streams applications to filter, enrich, and transform data. As we continue to expand our Kafka cluster, the new broker, connect workers, and stream processor servers are standardized as two-socket servers built with 28-core 2nd Gen Intel Xeon Platinum 6258R processors (56 physical cores per server) with 384 GB of memory.



Intel® Optane™ SSDs vs. 3D NAND: What is the Difference?

Intel® Optane™ SSDs are fundamentally different from 3D NAND. Data in NAND is grouped into blocks. As drive capacities grow, these blocks also grow, causing write delays and slowing the application performance. NAND also manages writes with complex wear-leveling and garbage-collection processes inside the SSD, creating additional overhead and increasing CPU wait time for data retrieval. Additionally, NAND cells can be written only a few thousand times before they no longer perform.

Intel® Optane™ memory media is a new technology built on a unique byte-addressable architecture similar to DRAM, yet non-volatile like NAND. Intel Optane memory media excels when accessing slow devices, networked storage, and in some caching or tiering models. In addition, the write speeds of Intel Optane memory media are similar to read speeds—well below 10 microseconds for a 4 kB storage block. This read/write balance makes it a great choice for demanding write-based applications, such as Apache Kafka's write log or Splunk's hot bucket.

Next Steps

As our CIP matures, we are planning new ways to take advantage of Confluent capabilities. Looking ahead, we are addressing the following:

- **Identify new cases.** We are actively identifying and including new security cases that can benefit from in-stream processing.
- **Increase I/O performance.** We plan to add Intel Optane SSDs as storage on our brokers. These SSDs, along with the upcoming tiered storage feature in the Confluent Platform, will help to significantly speed disk I/O and enhance message durability. Because clients consume and process data from Kafka in near real time, we plan to store up to 24 hours of data on Intel Optane SSDs and the rest on Intel 3D NAND SSDs. We also plan to put Intel Optane SSDs on the C3 server to help boost the performance and reduce start-up time of the C3 application.
- **Increase network speed.** Network speed is one of the most important factors in Kafka performance. Network bandwidth can become a bottleneck, impacting event throughput and latency. We are currently running our Kafka cluster on a 10 GbE network and plan to upgrade it to 40 GbE, 50 GbE, or 100 GbE to support the increasing workload.
- **Schema support and Confluent's ksqlDB.** ksqlDB is the purpose-built event-streaming database for stream-processing applications. Without schema, it can be difficult to work with ksqlDB, but recent JSON schema support helps us move closer to a complete streaming app with a few ksqlDB statements.

- **Machine learning.** The increasing need to detect malicious behavior in real time necessitates more than the ad hoc application of machine-learning techniques. We are working to develop in-stream machine-learning applications that can automatically normalize and join, filter, and enrich machine data (for example, syslog data) in-stream. For example, ksqlDB and Kafka Streams can be leveraged for feature engineering and for providing facts, features, and labels to model-generating algorithms or inference applications. Such model training and inference might also be implemented as a Kafka Streams application or ksqlDB User-Defined Function.

> "Kafka technology, combined with Confluent's enterprise features and high-performance Intel architecture, support our mission to make it safe for Intel to go fast."
>
> *–Brent Conran, Vice President and CISO, Intel*

Conclusion

The frequency and sophistication of cyber threats are constantly growing. At Intel, we invested in a modern, scalable CIP using Kafka and Splunk to help achieve this. We selected Confluent as our Kafka partner to help with the typical pain points and risks that accompany pure open source software deployments.

We immediately started to realize benefits of Kafka by achieving economies of scale. We can now acquire data once and consume it many times. Other benefits like reduction of technical debt by eliminating legacy point-to-point and custom connectors soon followed. And through data filtering, we have reduced the cost of ingest and storage in a variety of downstream systems.

The most transformational capability of Kafka for cybersecurity is in-stream processing. The ability to operate on data as it is produced helps security responders improve detection techniques and response times. It also enables us to develop and deploy more advanced techniques, such as machine-learning models that perform in-stream processing, which can identify threats in near real time. These advances are helping us achieve our mission to "make it safe for Intel to go fast." Using industry-leading technologies that provide a modern scalable architecture enables us to continue to transform well into the future.

Related Content

If you liked this paper, you may also be interested in these related stories:

- Transforming Intel's Security Posture with Innovations in Data Intelligence
- Advanced Persistent Threats: Hunting the One Percent
- Enabling Real-Time Processing of Massive Data Streams

For more information on Intel IT best practices, visit intel.com/IT.

IT@Intel

We connect IT professionals with their IT peers inside Intel. Our IT department solves some of today's most demanding and complex technology issues, and we want to share these lessons directly with our fellow IT professionals in an open peer-to-peer forum.

Our goal is simple: improve efficiency throughout the organization and enhance the business value of IT investments.

Follow us and join the conversation:
- Twitter
- #IntelIT
- LinkedIn
- IT Peer Network

Visit us today at intel.com/IT or contact your local Intel representative if you would like to learn more.

[1] Test Config: Test by Intel as of 6/30/2020. 1-node, 2x Intel® Xeon® Platinum 8280 processor, 28 cores, HT on, Turbo on, Total Memory 384 GB (12 slots/32 GB/2933 MHz), BIOS: SE5C6 20.86B.02.01.0010.010620200716 (ucode: 0x400002c), CentOS 8, 4.18.0-147.8.1.el8_1.x86_64, gcc (GCC) 8.3.1 20190507 (Red Hat 8.3.1-4), NIC X722 10G, Confluent Control Center 5.5, **Tunings:** Open JDK 11, Broker (java) Memory: 128 GB, Topic config: (60 Partitions, 3x replication, 2 min_insyn replicas, ack=1 [default],0,-1), Open files:16384, Max mmap:225000; **Config-1**: 4x 4 TB Intel® SSD DC P4510 (3D-NAND); **Config-2:** 4x 375 GB Intel® Optane™ SSD DC P4800X (3DXP).

[2] For performance comparisons, see intel.com/content/www/us/en/architecture-and-technology/optane-technology/faster-access-to-more-data-article-brief.html. For endurance information, see intel.com/content/www/us/en/architecture-and-technology/optane-technology/delivering-new-levels-of-endurance-article-brief.html.