Solution Reference Architecture
September 2022

Improving Splunk and Kafka Platforms with Cloud-Native Technologies

Intel® Select Solutions for Splunk and Kafka on Kubernetes use containers and S3-compliant storage to increase application performance and infrastructure utilization while simplifying the management of hybrid cloud environments

Contents

Authors:

Aleksander Kantak
Cloud Solutions Engineer, Intel

Murali Madhanagopal
Cloud Software Architect, Intel

Somu Rajarathinam
Technical Director, Pure Storage

Intel Contributors:
Mariusz Klonowski, Victor Colvard, Dennis Kwong, Elaine Rainbolt, Merritte Stidston, Jason Stark

Executive Summary

Data architects and administrators of modern analytic and streaming platforms like Splunk and Kafka continually look for ways to simplify the management of hybrid or multi-cloud platforms, while also scaling these platforms to meet the needs of their organizations. They are challenged with increasing data volumes and the need for faster insights and responses. Unfortunately, scaling often results in server sprawl, underutilized infrastructure resources and operational inefficiencies.

The release of Splunk Operator for Kubernetes and Confluent for Kubernetes, combined with Splunk SmartStore and Confluent Tiered Storage, offers new options for architectures designed with containers and S3-compatible storage. These new cloud-native technologies, running on Intel architecture and Pure Storage FlashBlade, can help improve application performance, increase infrastructure utilization and simplify the management of hybrid and multi-cloud environments.

Intel and Pure Storage architects designed a new reference architecture called Intel® Select Solutions for Splunk and Kafka on Kubernetes and conducted a proof of concept (PoC) to test the value of this reference architecture. Tests were run using Splunk Operator for Kubernetes and Confluent for Kubernetes with Intel IT's high-cardinality production data to demonstrate a real-world scenario.

In our PoC, a nine-node cluster reached a Splunk ingest rate of 886 MBps, while simultaneously completing 400 successful dense Splunk searches per minute, with an overall CPU utilization rate of 58%.[1] We also tested Splunk super-sparse searches and Splunk ingest from Kafka data stored locally versus data in Confluent Tiered Storage on FlashBlade, which exhibited remarkable results. The outcomes of this PoC informed the Intel Select Solutions for Splunk and Kafka on Kubernetes.

Keep reading to find out how to build a similar Splunk and Kafka platform that can provide the performance and resource utilization your organization needs to meet the demands of today's data-intensive workloads.

With our new reference architecture and PoC results, we identified four primary benefits for organizations deploying and managing large data platforms.

Solution Brief

Business Challenge

The ongoing digital transformation of virtually every industry means that modern enterprise workloads utilize massive amounts of structured and unstructured data. For applications like Splunk and Kafka, the explosion of data can be compounded by other issues. First, the traditional distributed scale-out model with direct-attached storage requires multiple copies of data to be stored, driving up storage needs even further. Second, many organizations are retaining their data for longer periods of time for security and/or compliance reasons. These trends create many challenges, including:

- **Server sprawl.** The distributed scale-out model, where data resides entirely in local storage on the compute node, is no longer practical and causes organizations to add both compute and storage resources when only storage is needed. It also drives up infrastructure and maintenance costs.
- **Under-utilized resources.** Adding compute resources just to get more storage results in CPU under-utilization, decreasing the return on investment in infrastructure.
- **Operational inefficiency.** Continually adding severs to a Splunk or Kafka cluster results in ongoing maintenance, including data rebalance, storage upgrades, firmware patches and other operational tasks.

Beyond the challenges presented by legacy architectures, organizations often have other challenges. Large organizations often have Splunk and Kaka platforms in both on-prem and multi-cloud environments. Managing the differences between these environments creates complexity for Splunk and Kafka administrators, architects and engineers.

Value of Intel® Select Solutions for Splunk and Kafka on Kubernetes

Many organizations understand the value of Kubernetes, which offers portability and flexibility and works with almost any type of container runtime. It has become the standard across organizations for running cloud-native applications; 69% of respondents from a recent Cloud-Native Computing Foundation (CNCF) survey reported using Kubernetes in production.[2] To support their customers' desire to deploy Kubernetes, Confluent developed Confluent for Kubernetes, and Splunk led the development of Splunk Operator for Kubernetes.

In addition, Splunk and Confluent have developed new storage capabilities: Splunk SmartStore and Confluent Tiered Storage, respectively. These capabilities use S3-compliant object storage to reduce the cost of massive data sets. In addition, organizations can maximize data availability by placing data in centralized S3 object storage, while reducing application storage requirements by storing a single copy of data that was moved to S3, relying on the S3 platform for data resiliency.

The cloud-native technologies underlying this reference architecture enable systems to quickly process the large amounts of data today's workloads demand; improve resource utilization and operational efficiency; and help simplify the deployment and management of Splunk and Kafka containers.

Solution Benefits

- **Meet business demands.** In our proof of concept (PoC), we built a platform that can process the vast amounts of data that modern machine learning and AI applications require, providing fast responses that can improve business outcomes.
- **Increase app performance and hardware utilization.** In our PoC, we used nine physical servers based on 3rd Generation Intel® Xeon® Scalable processors and Intel® Optane™ SSDs to support a single, combined Splunk and Kafka cluster consisting of 62 containers of Splunk search heads, Splunk indexers and Kafka brokers. We reached a peak Splunk ingest rate of 886 MBps, while simultaneously completing 400 successful dense Splunk searches per minute, with an average CPU utilization rate of 58%.[3]
- **Improve operational efficiency.** Splunk SmartStore with Pure Storage FlashBlade and Confluent Tiered Storage enable non-disruptive maintenance operations without the data evacuation and rebalance associated with traditional storage deployment models. Customers can save valuable time associated with software updates, patches and hardware refreshes.
- **Simplify management.** Cloud-native technologies like Kubernetes orchestrate and automate the deployment, configuring, scaling, monitoring and management of containerized hybrid cloud and multi-cloud environments.

Solution Architecture Highlights

We designed our reference architecture to take advantage of the previously mentioned new Splunk and Kafka products and technologies. We ran tests with a proof of concept (PoC) designed to assess Kafka and Splunk performance running on Kubernetes with servers based on high-performance Intel architecture and S3-compliant storage supported by Pure Storage FlashBlade.

Figure 1 illustrates the solution architecture at a high level. The critical software and hardware products and technologies included in this reference architecture are listed below:

- Splunk Enterprise with Splunk SmartStore
- Splunk Operator for Kubernetes
- Confluent Platform with Confluent Tiered Storage
- Confluent for Kubernetes
- S3-compliant Pure Storage FlashBlade
- Nine servers with:
  - 3rd Generation Intel® Xeon® Scalable processors
  - Intel® Optane™ P5800X SSDs
- Network architecture with Intel® Ethernet Adapters (see "Network Topology" and "FlashBlade Configuration Details" for more details about network setup)

Additional information about some of these components is provided in the "A Closer Look at Intel® Select Solutions for Splunk and Kafka on Kubernetes" section that follows.

What Are Intel® Select Solutions?

Intel Select Solutions are predefined, workload-optimized solutions designed to minimize the challenges of infrastructure evaluation and deployment. These solutions are validated by OEM/ODMs, certified by ISVs and verified by Intel.

All Intel Select Solutions are a tailored combination of Intel data center compute, memory, storage and network technologies that deliver predictable and compelling performance. Each solution offers assurance that the workload will perform as expected, if not better, which can save individual businesses from investing the resources that might otherwise be used to evaluate, select and purchase the hardware components to gain that assurance themselves.

intel SELECT SOLUTIONS

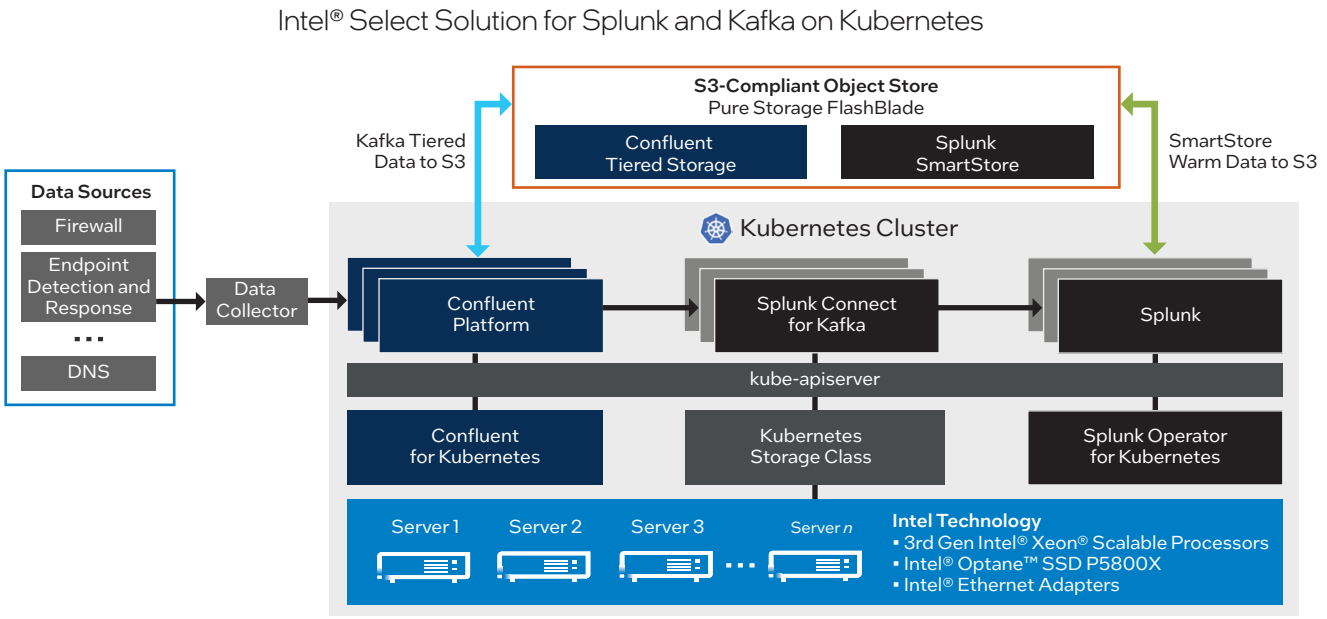

Figure 1. The solution reference architecture uses high-performance hardware and cloud-native software to help increase performance and improve hardware utilization and operational efficiency.

A Closer Look at Intel® Select Solutions for Splunk and Kafka on Kubernetes

The ability to run Splunk and Kafka on the same Kubernetes cluster connected to S3-compliant flash storage unleashes seamless scalability with an extraordinary amount of performance and resource utilization efficiency. The following sections describe some of the software innovations that make this possible.

Confluent for Kubernetes and Confluent Tiered Storage

Confluent for Kubernetes provides a cloud-native, infrastructure-as-code approach to deploying Kafka on Kubernetes. It goes beyond the open-source version of Kubernetes to provide a complete, declarative API to build a private cloud Kafka service. It automates the deployment of Confluent Platform and uses Kubernetes to enhance the platform's elasticity, ease of operations and resiliency for enterprises operating at any scale.

Confluent Tiered Storage architecture augments Kafka brokers with the S3 object store via FlashBlade, storing data on the FlashBlade instead of the local storage. Therefore, Kafka brokers contain significantly less state locally, making them more lightweight and rebalancing operations orders of magnitude faster. Tiered Storage simplifies the operation and scaling of the Kafka cluster and enables the cluster to scale efficiently to petabytes of data. With FlashBlade as the backend, Tiered Storage has the performance to make all Kafka data accessible for both streaming consumers and historical queries.

Splunk Operator for Kubernetes and Splunk SmartStore

The Splunk Operator for Kubernetes simplifies the deployment of Splunk Enterprise in a cloud-native environment that uses containers. The Operator simplifies the scaling and management of Splunk Enterprise by automating administrative workflows using Kubernetes best practices.

Splunk SmartStore is an indexer capability that provides a way to use remote object stores to store indexed data. SmartStore makes it easier for organizations to retain data for a longer period of time. Using FlashBlade as the high-performance remote object store, SmartStore holds the single master copy of the warm/cold data. At the same time, a cache manager on the indexer maintains the recently accessed data. The cache manager manages data movement between the indexer and the remote storage tier. The data availability and fidelity functions are offloaded to FlashBlade, which offers N+2 redundancy.[4]

Remote Object Storage Capabilities

Pure Storage FlashBlade is a scale-out, all-flash file and object storage system that is designed to consolidate complete data silos while accelerating real-time insights from machine data using applications such as Splunk and Kafka. FlashBlade's ability to scale performance and capacity is based on five key innovations:

- An all-flash architecture with integrated non-volatile random-access memory (NVRAM).
- A unified network that supports IPv4 and IPv6 client access over Ethernet links up to 100 Gb/s.
- Purity//FB storage operating system minimizes workload balancing problems by distributing all client operation requests evenly among blades.
- A common media architectural design for files and objects supports concurrent access to files using a variety of protocols such as NFSv3, NFS over HTTP and SMB and objects via S3.
- Ease of use enabled by autonomously performing routine administrative tasks, self-tuning and providing system alerts when components fail.

A complete FlashBlade system configuration consists of up to 10 self-contained rack-mounted servers. A single 4U chassis FlashBlade can host up to 15 blades and a full FlashBlade system configuration can scale up to 10 chassis (150 blades), potentially representing years of data for even higher ingest systems. Each blade assembly is a self-contained compute module equipped with processors, communication interfaces and either 17 TB or 52 TB of flash memory for persistent data storage. Figure 2 shows how the reference architecture uses Splunk SmartStore and FlashBlade.

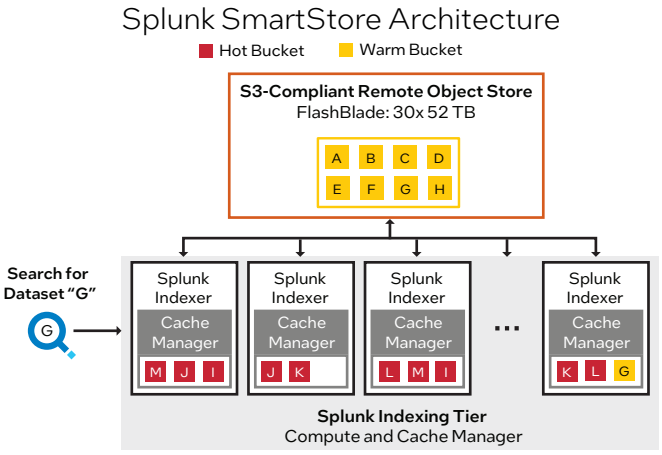

Figure 2. Splunk SmartStore using FlashBlade for the remote object store.

Proof of Concept Testing Process and Results

The following tests were performed in our PoC:

- **Test #1**: Test Splunk ingest rates with simultaneous dense searches running on bare-metal servers and then on the same servers running Splunk and Kafka on Kubernetes.
- **Test #2**: Test Splunk ingest rate by reading input data from the Kafka local storage versus the input data from the Confluent Tiered Storage hosted on FlashBlade.
- **Test #3**: Test time required to run Splunk super-sparse searches accessing data from Splunk SmartStore residing on FlashBlade.

For all the tests, we used Intel IT's real-world high-cardinality production data from sources such as DNS, Endpoint Detection and Response (EDR) and Firewall, which were collected into Kafka and ingested into Splunk through Splunk Connect for Kafka.

Test #1: Application Performance and Infrastructure Utilization

In this test, we compared the performance of a bare-metal Splunk and Kafka deployment to a Kubernetes deployment. The test consisted of reading data from four Kafka topics and ingesting that data into Splunk, while dense searches were scheduled to run every minute.

Bare-Metal Performance

We started with a bare-metal test using nine physical servers. Five nodes served as Splunk indexers, three nodes as Kafka brokers and one node served as a Splunk search head. With this bare-metal cluster, the peak ingest rate was 301 MBps, while simultaneously finishing 90 successful Splunk dense searches per minute (60 in cache, 30 from FlashBlade), with an average CPU utilization of 12%. The average search runtime for the Splunk dense search was 22 seconds.

Addition of Kubernetes

Next, we deployed Splunk Operator for Kubernetes and Confluent for Kubernetes on the same nine-node cluster. Kubernetes spawned 62 containers: 35 indexers, 18 Kafka brokers and nine search heads. With this setup, we reached a peak Splunk ingest rate of 886 MBps, while simultaneously finishing 400 successful Splunk dense searches per minute (300 in cache, 100 from FlashBlade), with an average CPU utilization of 58%. The average search runtime for the Splunk dense search was 16 seconds—a 27% decrease from the Splunk average search time on the bare-metal cluster. Figure 3 illustrates the improved CPU utilization gained from containerization using Kubernetes. Figure 4 shows the high performance enabled by the reference architecture.

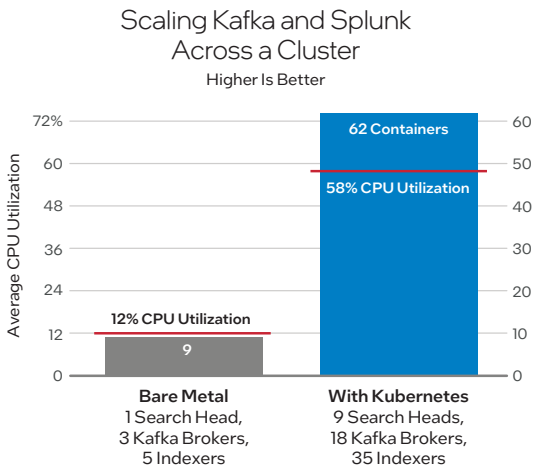

Figure 3. Deployment of the Splunk Operator for Kubernetes and Confluent for Kubernetes enabled 62 Splunk and Kafka containers on the nine physical servers in the PoC cluster.

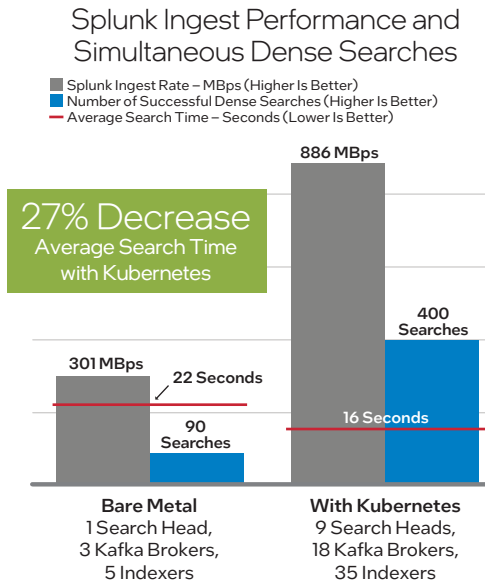

Figure 4. Running Splunk Operator for Kubernetes and Confluent for Kubernetes enabled up to 2.9X higher ingest rate, up to 4x more successful dense searches, and a 27% reduction in average Splunk search time, compared to the bare-metal cluster.

Test #2: Data Ingest from Kafka Local Storage versus Confluent Tiered Storage

Kafka's two key functions in event streaming are producer (ingest) and consumer (search/read). In the classic Kafka setup, the produced data is maintained at the broker's local storage, but with Tiered Storage, Confluent offloads the data from the Tiered Storage to the object store and enables infinite retention. If any consumer is looking for data that is not in the local storage, the data would be downloaded from the object storage.

To compare the consumer/download performance, we started the Splunk Connect workers for Kafka after one hour of data ingestion into Kafka with all data on the local SSD storage. The Connect workers read the data from Kafka and forwarded it to the Splunk indexers, where we measured the ingest throughput and elapsed time to load all the unconsumed events. During this time, Kafka read the data from the local SSD storage, and Splunk was also writing the hot buckets into the local SSD storage that hosts the hot tier.

We repeated the same test when the topic was enabled with Tiered Storage by starting the Splunk Connect workers for Kafka, which initially read the data out of FlashBlade and later from the local SSD storage for the last 15 minutes. We then measured the ingest throughput and the elapsed time to load all the unconsumed events.

As shown in Figure 5, there is no reduction in the Kafka consumer performance when the broker data is hosted on Tiered Storage on FlashBlade. This reaffirms that offloading Kafka data to the object store, FlashBlade, gives not only similar performance for consumers but also the added benefit of longer retention.

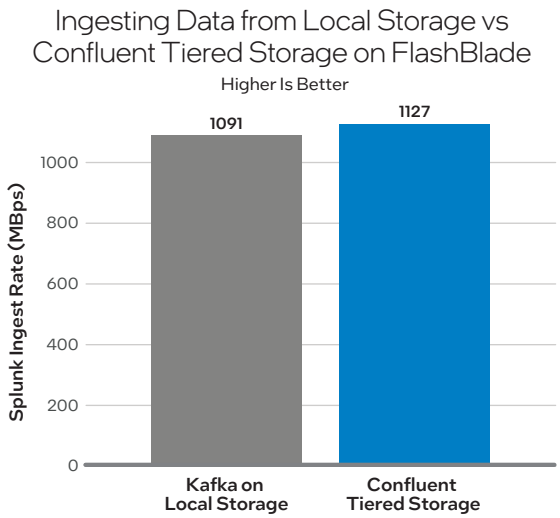

Figure 5. Using Confluent Tiered Storage with FlashBlade enables longer data retention while maintaining (or even improving) the ingest rate.

Test #3: Splunk Super-Sparse Searches in Splunk SmartStore

When data is in the cache, Splunk SmartStore searches are expected to be similar to non-SmartStore searches. When data is not in the cache, search times are dependent on the amount of data to be downloaded from the remote object store to the cache. Hence, searches involving rarely accessed data or data covering longer time periods can have longer response times than experienced with non-SmartStore indexes. However, FlashBlade accelerates the download time considerably in comparison to any other "cheap-and-deep" object storage available today.[4]

To demonstrate FlashBlade's ability to accelerate downloads, we tested the performance of a super-sparse search (the equivalent of finding a needle in a haystack); the response time of this type of search is generally tied to I/O performance. The search was initially performed against the data in the Splunk cache to measure the resulting event counts. The search returned 64 events out of several billion events. Following this, the entire cache was evicted from all the indexers, and the same super-sparse search was issued again, which downloaded all the required data from FlashBlade into the cache to perform the search. We discovered that FlashBlade supported a download of 376 GB in just 84 seconds with a maximum download throughput of 19 GBps (see Table 1).

Table 1. Results from Super-Sparse Search

Results	
Downloaded Buckets	376 GB
Elapsed Time	84 seconds
Average Download Throughput	4.45 GBps
Maximum Download Throughput	19 GBps

A super-sparse search downloading
376 GB in 84 Seconds

Configuration Summary

Introduction

The previous pages provided a high-level discussion of the business value provided by Intel Select Solutions for Splunk and Kafka on Kubernetes, the technologies used in the solution and the performance and scalability that can be expected. This section provides more detail about the Intel technologies used in the reference design and the bill of materials for building the solution.

Intel Select Solutions for Splunk and Kafka on Kubernetes Design

The following tables describe the required components needed to build this solution. Customers must use firmware with the latest microcode. Tables 2, 3 and 4 detail the key components of our reference architecture and PoC. The selection of software, compute, network, and storage components was essential to achieving the performance gains observed.

Table 2. Key Server Components

Component	Description
CPU	2x Intel® Xeon® Platinum 8360Y (36 cores, 2.4 GHz)
Memory	16x 32 GB DDR4 @ 3200 MT/s
Storage (Cache Tier)	1x Intel® Optane™ SSD P5800x (1.6 TB)
Storage (Capacity Tier)	1x SSD DC P4510 (4 TB)
Boot Drive	1x SSD D3-S4610 (960 GB)
Network	Intel® Ethernet Network Adapter E810-XXVDA2 (25 GbE)

Table 3. Software Components

Software	Version
Kubernetes	1.23.0
Splunk Operator for Kubernetes	1.0.1
Splunk Enterprise	8.2.0
Splunk Connect for Kafka	2.0.2
Confluent for Kubernetes	2.2.0
Confluent Platform	7.0.1 using Apache Kafka 3.0.0

Table 4. S3 Object Storage Components

Pure Storage FlashBlade	Description
FlashBlades	30x 52 TB blades
Capacity	1560 TB raw 1440 TB usable (with no data reduction)
Connectivity	4x 100 Gb/s Ethernet (data) 2x 1 Gb/s redundant Ethernet (management port)
Physical	10U (4U per chassis, 1U per XFM)
Software	Purity//FB 3.1.10

Network Topology

Figure 6 illustrates the network layout that is used in this reference design.

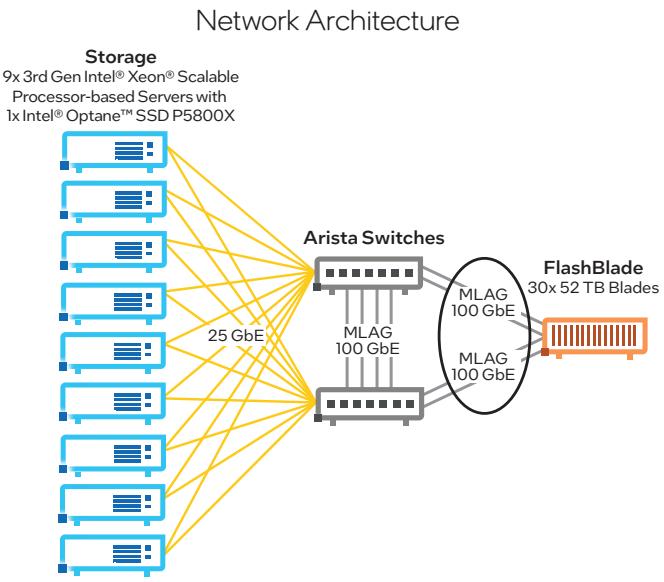

Figure 6. Network topology used in the Intel® Select Solutions for Splunk and Kafka on Kubernetes.

Local Storage Details

The local NAND and Intel Optane SSDs on the servers were used to provision the hot/cache tier of the Splunk indexers and the Kafka brokers using the Local Path Provisioner tool, which uses the Local Persistent Volume feature of Kubernetes and creates the hostPath-based persistent volume on the nodes automatically.

FlashBlade Configuration Details

The FlashBlade chassis are interconnected by high-speed links to two external fabric modules (XFMs). Two onboard fabric modules are at the rear of each chassis to interconnect the blades, other chassis and clients using TCP/IP over high-speed Ethernet. Both XFMs are interconnected, and each contains a control processor and Ethernet switch ASIC. For reliability, each chassis is equipped with redundant power supplies and cooling fans. The front of each chassis holds up to 15 blades for processing data operations and storage, and can support more than 1.5 million NFS IOPS, or up to 15 GB/s of bandwidth on a single chassis with 15 blades on a 3:1 compressible dataset.[5]

Key Technologies

Several Intel products were utilized to help improve performance.

3rd Generation Intel Xeon Scalable Processors

Intel's latest processors for data center workloads are 3rd Gen Intel Xeon Scalable processors. They are packed with performance- and security-enhancing features, including the following:

- Enhanced per-core performance, with up to 40 cores in a standard socket
- Enhanced memory performance with support for up to 3200 MT/s DIMMs (2 DIMMs per channel)
- Increased memory capacity with up to eight channels
- Faster inter-node connections with three Intel® Ultra Path Interconnect links at 11.2 GT/s
- More, faster I/O with PCI Express 4 and up to 64 lanes (per socket) at 16 GT/s

Intel Optane SSDs

Intel Optane SSD P5800X with next-generation Intel Optane storage media and advanced controller delivers "no-compromises" I/O performance—read or write. It also has high endurance, providing unprecedented value over legacy storage in the accelerating world of intelligent data. Intel Optane SSD P5800X delivers 4x greater random 4K mixed read/write IOPS and 67 percent higher endurance compared to the previous-generation Intel Optane SSD DC P4800X, which uses PCIe gen 3.[6]

Intel Ethernet 800 Series

The Intel Ethernet 800 Series is the next evolution in Intel's line of Ethernet products. Compared to the Intel Ethernet 700 Series, the 800 Series offers higher bandwidth due to the use of PCIe 4.0 and 50 Gb PAM4 SerDes. It also improves application efficiency with Application Device Queues and enhanced Dynamic Device Personalization. The 800 Series is versatile, offering 2x 100/50/25/10 GbE, 4x 25/10 GbE or 8x 10 GbE connectivity. It also supports RDMA for both iWARP and RoCE v2, which gives enterprises a choice when designing their hyperconverged networks.

Reference Design and PoC Key Learnings

While building out our reference architecture, the following key learnings came to light:

- To achieve optimal performance and scaling, it's recommended to follow best-practices guides for reference architecture components such as Splunk and Kafka.
- In particular, it is advisable to verify that you are using the correct version of software, firmware and OS; sometimes the recommended version is the latest version, but not always.
- Because these cloud-native technologies are relatively new, plan on performing trial and error with configurations prior to production implementation.
- Verify that you have set up monitoring tools for the different layers of software and hardware, so you can compare results over time.
- Utilize high-core-count 3rd Gen Intel Xeon Platinum or Gold processors to provide high-performance compute.

Conclusion

The demand for data as well as the business opportunities from large data sets has never been greater. But the challenges of harnessing and managing data continue to plague organizations of all sizes, and in every industry. The release of Splunk Operator for Kubernetes and Confluent for Kubernetes, combined with Splunk SmartStore and Confluent Tiered Storage, offer new data architectures designed with containers and S3-compatible storage. In our PoC, we demonstrated how these new cloud-native technologies, running on Intel architecture and Pure Storage FlashBlade, can help improve application performance and operational efficiency, increase infrastructure utilization and simplify the management of hybrid and multi-cloud environments.

Learn More

- Transforming Intel's Security Posture with Innovations in Data Intelligence white paper
- Building a Modern, Scalable Cyber Intelligence Platform with Apache Kafka white paper
- Deliver Insights Faster with Intel® Select Solutions for Containerized Splunk solution brief
- Now Available: Pure FlashBlade for Confluent Tiered Storage blog
- Splunk SmartStore on Pure FlashBlade white paper

Find the solution that is right for your organization. Contact your Intel representative.

Revision History

Document Number	Revision Number	Description	Date
	1.0	First Release	June 2022

[1] See intel.com/performanceindex for workloads and configurations. Results may vary. Specifically, refer to "DTCC003 Maximizing Performance, Scalability and Operational Efficiency with Kubernetes and Splunk SmartStore" at https://edc.intel.com/content/www/us/en/products/performance/benchmarks/vision-2022

[2] ContainIQ, "26 Kubernetes Statistics to Reference," https://www.containiq.com/post/kubernetes-statistics

[3] See endnote 1.

[4] Pure Storage, "Splunk SmartStote on FlashBlade," https://support.purestorage.com/Solutions/Splunk/Splunk_Reference/Splunk_SmartStore_on_FlashBlade

[5] "Pure Storage FlashBlade Data Sheet," https://www.purestorage.com/products/file-and-object/flashblade/data-sheet.html

[6] See [2] and [15] at https://edc.intel.com/content/www/us/en/products/performance/benchmarks/intel-optane-ssd-p5800x-series/