KINNEY GROUP

DIAMANTI

Splunk Reference Architecture

Deploying Splunk on the Diamanti Platform

Executive Summary

Diamanti and Kinney Group have collaborated to create best of class reference architectures for Splunk Enterprise and Splunk Enterprise Security. Deploy Splunk faster with less risk, better performance, and lower cost.

The Diamanti Spektra + Splunk Reference Design will demonstrate the benefits of deploying Splunk onto the Diamanti platform as opposed to traditional cloud deployments. Utilizing Diamanti's advanced storage capabilities and the ease of deployment that comes with Kubernetes, this Reference Design will highlight the performance, cost benefits, and time savings of deploying Splunk on the Diamanti platform.

This reference architecture demonstrates that deploying Splunk Enterprise and Splunk Enterprise Security on Diamanti's full-stack solution outperforms a similarly built AWS infrastructure. Further, the Diamanti solution is simpler, faster, and has a lower total cost of ownership than a comparable AWS environment.

GOALS AND OBJECTIVES

This reference architecture aims to validate the performance and compatibility of utilizing Splunk Enterprise and Splunk Enterprise Security on the Diamanti's platform.

The key objectives for this reference architecture are to:

- Prove the power behind automation supporting the deployment and configuration of Splunk Enterprise and Splunk Enterprise Security on Kubernetes.
- Demonstrate how a methodology that supports deploying Splunk on Diamanti's hardware using a Splunk Validated Architecture (SVA) and best practices outperforms a comparable system on AWS.
- Highlight how the Diamanti's superior capacity and scalability can provide substantial savings for Splunk deployments.

The Diamanti + Splunk Reference Design will demonstrate the above goals and objectives by deploying Splunk Enterprise and Splunk Enterprise Security onto the Diamanti platform as opposed to traditional cloud or on-premises deployments. Utilizing Diamanti's advanced storage capabilities and the ease of deployment that is native to Kubernetes, this Reference Design will highlight the time savings, performance, and cost benefits of deploying Splunk on the Diamanti platform. The Diamanti platform is compatible with Splunk Validated Architectures (SVAs).

Introduction

CHALLENGES WITH OPERATING SPLUNK AT SCALE

Splunk helps IT, DevOps and Security teams reduce downtime and strengthen cybersecurity. However, as the amount of data continues to grow, enterprises need to scale their Splunk environment in a cost-efficient way across a variety of environments. This introduces two primary challenges:

- **Deployment complexity** – Standing up a new Splunk Enterprise and Splunk Enterprise Security environment has improved over the years, but it can still take several weeks. As enterprises move to hybrid cloud models, they need faster ways to deploy, patch and upgrade their Splunk environments either on-premises or in the public cloud.
- **Performance** – For many enterprises running Splunk Enterprise, the limitation on data ingestion is tied to indexing latency. Without enough capacity, enterprises risk skipped searches which in turn increases the risk of data loss and impacts the accuracy of the analytic results. For many enterprises, that means oversizing a cluster to support growing data needs.

CONTAINERIZING SPLUNK FOR SIMPLE DEPLOYMENT

Since the debut of Docker containers in 2013, the industry has been rapidly standardizing scale-out applications on containers and Kubernetes. Splunk Enterprise is one such application that can benefit from its many advantages:

- **Lower Total Cost of Ownership (TCO)**: A containerized deployment can lead to a smaller footprint and more efficient resource utilization.
- **Faster Time-to-Value**: Containers make the deployment of applications easier and faster. Organizations can test, patch and upgrade environments rapidly, using a declarative approach to application deployment.
- **Increased agility and flexibility**: Containerized applications are extremely portable and can be consistently deployed both on-premises and in the cloud.

Combined, these advantages all integrate into something that's very desirable in any Splunk environment — **accelerated time to insight**. Containers make Splunk simple to deploy and update, which is why Splunk has, since 2019, accelerated the containerization of Splunk through both development efforts and official support.

DEPLOY SPLUNK ON CLOUD-NATIVE INFRASTRUCTURE

Once containerized, Splunk Enterprise and Splunk Enterprise Security can be deployed in Kubernetes environments that can run anywhere. However, as a data-intensive application, running a containerized Splunk environment on legacy infrastructure soon runs into performance issues.

The Diamanti platform is purpose-built for Kubernetes. Using Diamanti's cloud-native, distributed storage architecture and NVMe based storage, Splunk can generate extremely fast indexing with low latency and consistent high performance. That means enterprises can process more data in a shorter amount of time, with a smaller server footprint.

For many organizations, running Splunk on Diamanti means being able to drive actionable measures in near real-time, enabling faster insights, and reducing TCO.

Solution Design

REFERENCE ARCHITECTURE DESIGN PRINCIPLES

The guiding principles for implementing this reference architecture are:

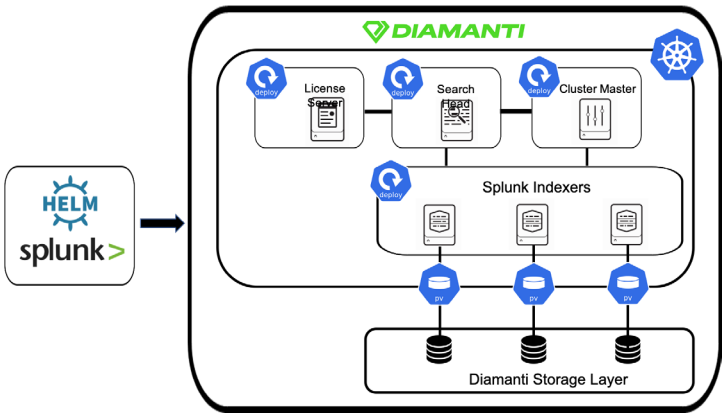
Simple Integrate Helm charts to deploy Splunk in a simple and repeatable process, reducing time and complexity.

Efficient Utilize the Diamanti Spektra platform to optimize available resources, when compared to the same Splunk architecture on public clouds.

Performant Enable customers to easily scale up their Splunk infrastructure based on daily data volume needs and achieve high performance

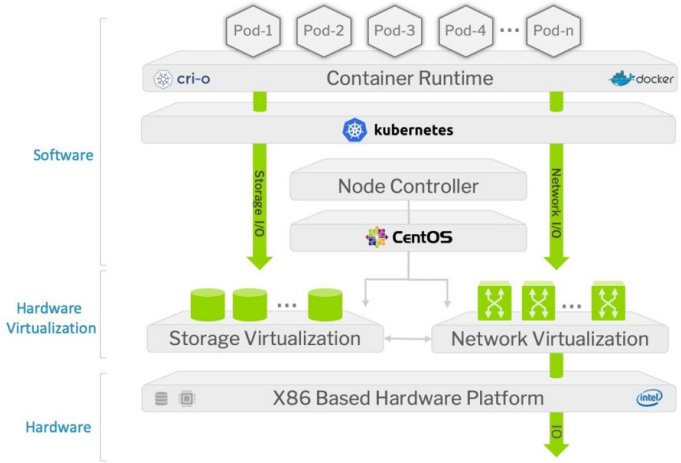
Cost-effective Do more with less. Reduce overall server footprint and related cost of ownership in utilizing Diamanti's unique architecture and accelerated scalability.

DESIGN TOPOLOGY



Splunk Ansible in conjunction with Helm were used to deploy the Splunk Reference Architecture. Splunk Ansible and Helm were selected due to their simplistic nature of deploying applications within a shortened time frame. In many cases, Splunk Administrators will be looking for a solution from Splunk. Splunk Operator was officially introduced in 2019. The Splunk Operator for Kubernetes (SOK) makes it easy for Splunk Administrators to deploy and operate Enterprise deployments in a Kubernetes infrastructure. However, Splunk operator only supports a small subset of Splunk configurations. Splunk Ansible and Helm can support a full range of Splunk Validated Architectures at scale. Further, there is a larger community utilizing Helm charts for Splunk deployments. The use of Helm charts helps manage complexity, ease updates, and apply standards to the deployment.

DIAMANTI STACK



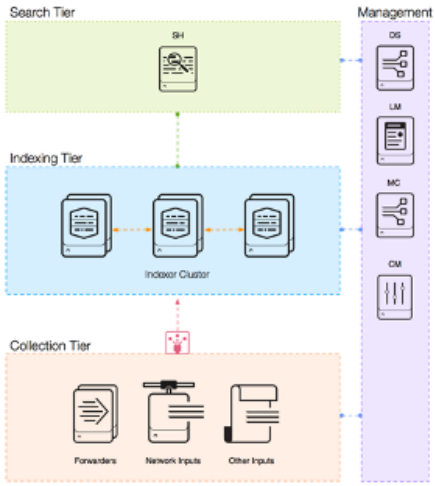
Diamanti is solving the challenge of container-based hybrid clouds with turnkey solutions for managing Kubernetes at any scale. Diamanti's Spektra platform propels enterprises to break from legacy architectures and rapidly adopt and expand Kubernetes on-premises and in the cloud, with security, high availability and resilience built in. With Diamanti, you can deploy a complete enterprise-class Splunk container stack based on a Splunk Validated Architecture in minutes, complete with compute, networking, storage, and fully-integrated Docker and Kubernetes.

SPLUNK ARCHITECTURE

The architecture chosen for this solution comes from Splunk's Validated Architectures (SVA). For more information about the architecture chosen, please see:

https://www.splunk.com/pdfs/technical-briefs/splunk-validated-architectures.pdf

The selected architecture is based on Splunk's C1 design for its simple and scalable nature. The design makes use of a single search head to run Splunk Enterprise Security, and multiple indexers in a cluster for data resiliency. The Splunk C1 design was chosen as it is the most frequently used SVA out of all reference design documents.



The basic topology consists of a Search Head running out of the box, untuned Enterprise Security; a clustered indexing tier with a non-indexing Master Node; a License Master; and one (1) or more Universal Forwarders generating sample data to send to the index cluster. Because this environment operates on Kubernetes and uses the Splunk support Ansible playbooks, all apps are deployed when the containers are built, and thus no Deployment Server is used.

CONTAINER CONFIGURATIONS

Server	Limits (CPU/RAM)	Requests (CPU/RAM)
Search Head	16 CPU/12 GB	8 CPU/6 GB
Cluster Master	4 CPU/8 GB	2 CPU/4 GB
Indexers	12 CPU/12 GB	6 CPU/6 GB
License Master	2 CPU/4 GB	2 CPU/2 GB
Universal Forwarders	4 CPU/8 GB	2 CPU/4 GB

All Splunk components are running Splunk Version 8.0.3 (build a6754d844abf) and are deployed and configured using the Splunk supported Ansible playbooks. Using the Helm charts and playbooks enabled testing on both the Diamanti platform and AWS to be configured rapidly compared to traditional installation methods. **It took only a matter of hours versus several days.**

To validate the architecture, benchmark testing was performed comparing the same deployment of Splunk Enterprise on the Diamanti platform and on AWS EKS – the Kubernetes service that runs on EC2 instances.

HARDWARE USED

In Diamanti's hardware lab environment, testing consisted of three (3) D20 nodes in a single cluster. The Diamanti D20 family is a modern hyperconverged platform consisting of varying configurations of Intel CPUs, memory and NVMe storage.

AWS EC2 instances were chosen to closely resemble the Diamanti Environment in terms of resources and node count. An EC2 instance is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications on the Amazon Web Services (AWS) infrastructure. For storage, we selected a specialized EBS class with 500G drives and 10K provisioned IOPS per node.

AWS EC2 instances were chosen to closely resemble the Diamanti environment in terms of CPU. For storage, we selected an EBS class with 500G drives and 10K provisioned IOPS per node.

AWS AND KUBERNETES NODE COMPARISON

Per Node	CPUs	RAM
Diamanti	40	128 GB
AWS c5.9xlarge	36	72 GB

AWS provisioned IOPS were allocated for the EC2 instances. Provisioned IOPS are a new EBS volume type designed to deliver predictable, high performance for I/O intensive workloads.

SOFTWARE USED

Splunk components were loaded and tested with robust load generation scripts for data sets including firewall, proxy, end point, network, operating system, and web server logs. Splunk Enterprise Security was configured to use more than 25 correlation searches, numerous saved searches, and manually tested with ad hoc searches to simulate a typical enterprise customer. Bonnie++ software and the Linux "dd" command were used to add artificial IOPS stress on systems during testing.

For more information on all software used on Splunk systems see Appendix B.

Performance Tests

TESTING PROCEDURES

Testing is designed to mimic real-world customer environments with a single index cluster and Enterprise Security running on a single search head. The integration of Kubernetes allows for easy reset of the entire testing environment to ensure that previous tests have no impact on current or future tests.

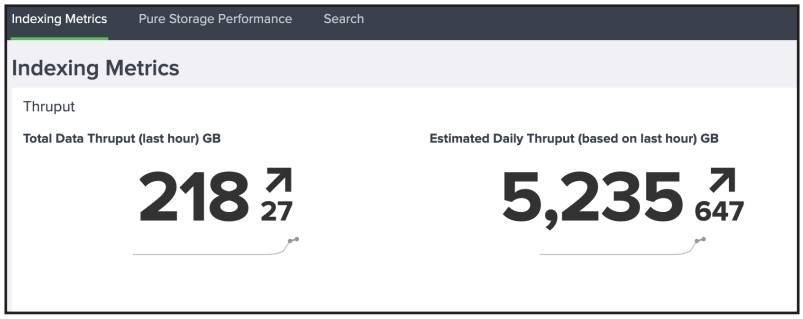
1. Deploy a Splunk environment with the appropriate number of clustered indexers.
2. Validate the deployment completed correctly through the Splunk Monitoring Console, Indexer Clustering interface, and Kubernetes kubectrl.
3. Deploy the appropriate number of Universal Forwarders to generate data.
4. Utilize the KGI Metrics App, Splunk Monitoring Console, and other management dashboards (as appropriate) to measure the health of data ingestion queues, search scheduler skip ratio, IOPS, memory and CPU utilization.
5. Manually stress IOPS on individual containers and nodes using Bonnie++ and Linux “dd” command.

DATA INGEST AND MANAGEMENT

Load generation in both environments scaled from 250 GB to 2+ TB of daily ingest. The following notes and graphics show the impact of load of each environment at the high ingestion rate. Daily ingest rates below 2+ TB showed no significant impact on either environment unless manual IOPS generation was used to simulate production environments. At higher loads, each data generation device was capable of generating 1+ TB per day of daily ingest.

Diamanti Results

At peak load, the Diamanti cluster sustained 5+ Terabytes of daily ingest.



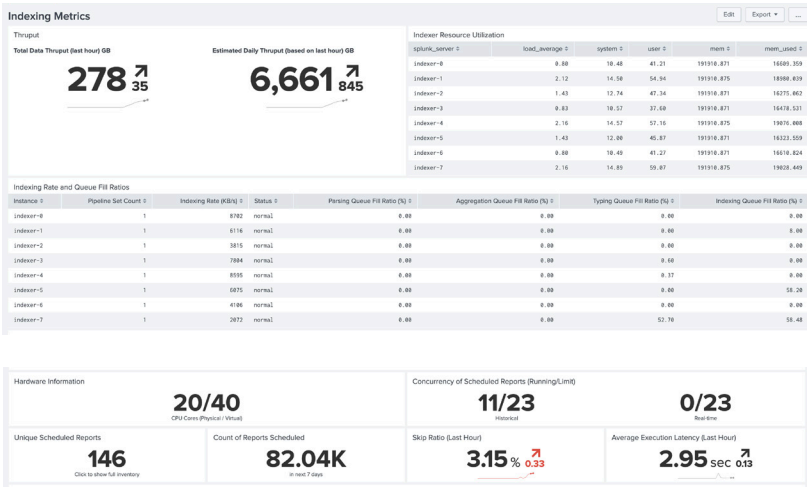
The Diamanti system gained no impact to indexer queues, even at load.

Parsing Queue Fill Ratio (%)	Aggregation Queue Fill Ratio (%)	Typing Queue Fill Ratio (%)	Indexing Queue Fill Ratio (%)
0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.04
0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00

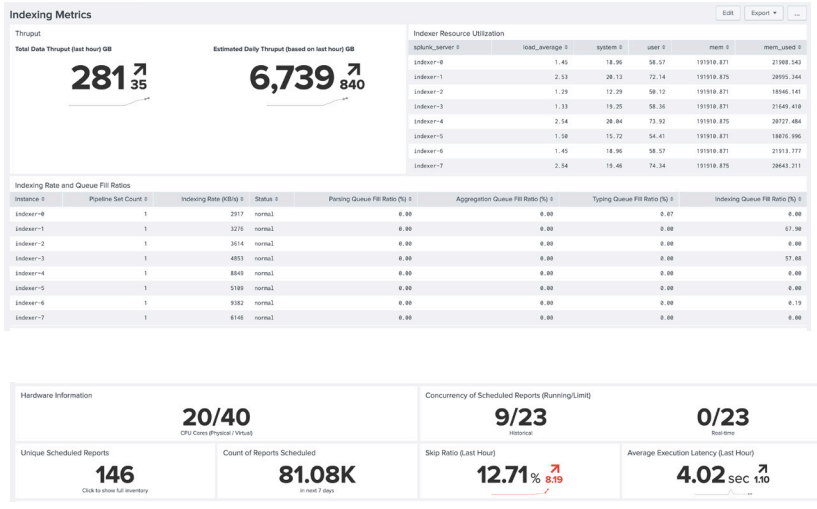
In summary, the sustainable load met 5 TB, which was 2.5x the Splunk recommended rate for hardware configuration. CPU/RAM/Network utilization was normal for load. This sustained high performance allows customers to avoid some of the common pitfalls of scaling Splunk. There was little to no indexer queue latency or search execution latency. Finally, there were no skipped searches even at the highest load.

Diamanti Stress Test

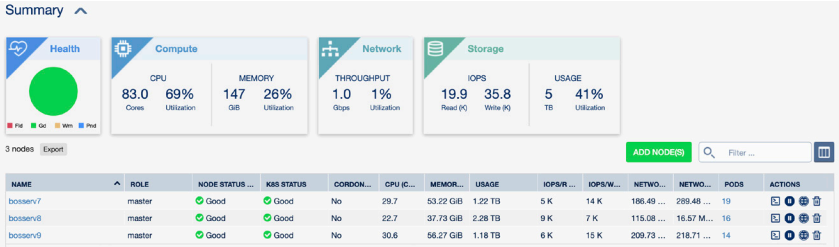
Diamanti sustained loads of 5+ Terabytes of daily data ingest with no impact to indexing or search performance. During subsequent stress testing the Diamanti system was able to reach loads of 6.6 Terabytes for sustained periods with only intermittent indexer queueing and slight search performance impact.



Increasing load past 6.6 TB strained the system and performance degraded rapidly about 6.7 TB.

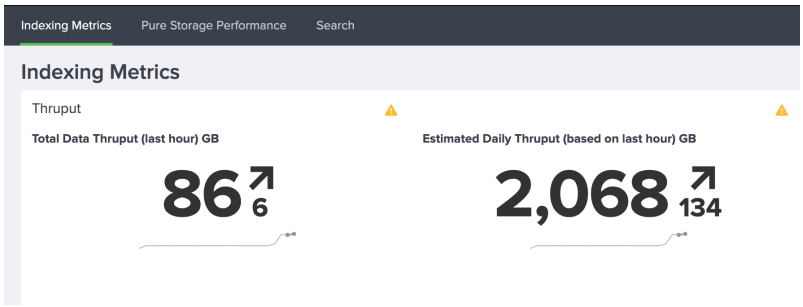


The Diamanti cluster easily handled the required IOPS for these level of data ingest, but appeared to become CPU bound at peak data ingest and search usage.



AWS Results

The AWS system ingestion rate capped at ~2TB of daily ingest.



Even at this lower ingest rate, the AWS indexers started to have indexer queue issues.

Parsing Queue Fill Ratio (%)	Aggregation Queue Fill Ratio (%)	Typing Queue Fill Ratio (%)	Indexing Queue Fill Ratio (%)
0.00	0.00	0.11	0.09
0.00	0.00	0.00	0.00
19.57	99.92	100.00	100.00
0.00	0.43	91.49	74.87
0.00	8.22	99.97	77.38
0.00	1.67	99.98	99.96

The biggest problems with Splunk solutions at scale are indexer latency, skipped searches, and slow searches. Indexer latency can result in data indexing too slow to be useful in real time search and investigations. This data is critical in any security solution to support Incident Response and Threat Hunting use cases. Skipped searches can result in critical analytics failing to run meaning time sensitive security alerts may never reach the SOC. Unfortunately many back end and all user related functions are searches. Dashboards are based on searches, reports are based on searches, alerts are based on searches, and of course searches are searches. When searches are slow, users suffer. Users might be SOC analysts, threat hunters, IT Operations, or even business analysts. These roles are mission critical to most organizations. Simply put - **Splunk search needs to work and perform for the system to provide any value**. In summary, the performance impact of sustained load resulted in skipped searches and an almost unusable user experience on the AWS cluster. Search performance and user experience on the Diamanti cluster remained stable throughout testing. Skipped searches and poor search performance over 80% during heavy load.

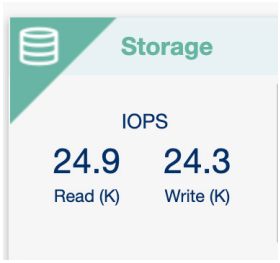
RESILIENCY

Manual disk interaction through Bonnie++ software and the Linux “dd” command was introduced to each environment in a scaled fashion.

Diamanti Results

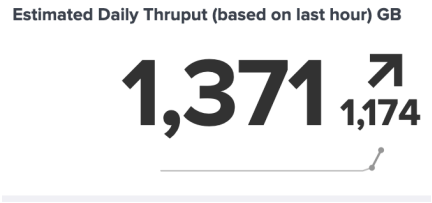
The Diamanti cluster saw a spike in IOPS, but was not otherwise impacted at the node or container level.

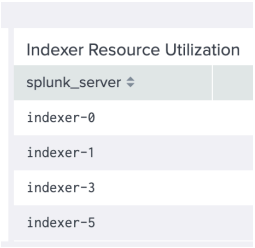
NODE STATUS ...	K8S STATUS	CORDON...	CPU (C...	MEMOR...	USAGE	IOPS/R ...	IOPS/W...	NETWO...	NETWO...	PODS
Good	Good	No	19.9	67.45 GiB	805.62 GB	5 K	9 K	101.05 ...	148.06 ...	16
Good	Good	No	34.6	74.79 GiB	1.52 TB	18 K	14 K	302.56 ...	234.07 ...	13
Good	Good	No	12.4	46.74 GiB	741.13 GB	2 K	2 K	58.92 M...	46.98 M...	12



AWS Results

Under manual stress testing the AWS cluster immediately dropped in overall capacity. Ultimately, the Cluster evicted multiple containers including the Splunk indexer.





The AWS container evictions resulted in nonrecoverable systems. This would result in data loss at the indexer replication level. These failures were later replicated on systems with less load showing the inherent risk of running Splunk with shared disk systems.

SCALABILITY & COST EFFECIENCY

The Diamanti cluster was able to scale to over five [5] TB of daily ingest without any noticeable impact to performance. This is over 2.5x the recommended rate on a standard deployment. The ability to oversubscribe the underlying Diamanti architecture provides substantial costs savings. Diamanti provides subscription and perpetual based pricing.

The AWS cluster scaled to the recommended rate for the architecture, but was unable to handle spikes in IOPS on the Kubernetes cluster. The operational costs for the AWS infrastructure are approximated here so a total cost of ownership comparison can be understood.

AWS	Cost per year
Reserved Linux Server Cost	$25,333.92
Storage cost *(@ 2 TB/ day for 30 days)*	$92,160
Provisioned IOPS *(10,000/ server)*	$23,400

The estimated AWS infrastructure annual cost is more than $141,000 which doesn't include Splunk licensing, hidden AWS costs (data transfers, elastic IP addresses, etc.), and other associated fees. This is a substantial annual recurring cost which results in a "total infrastructure cost of ownership" (TCO) of approximately $70/GB ingested.

Subscription based pricing for the Diamanti hardware used in this reference architecture, with no hidden costs and easily capable of handling 4+ TB of daily ingest has a "total infrastructure cost of ownership" of approximately $26/GB ingested. Extending the number of nodes, data retention period, and/or increasing the provisioned IOPS on AWS will further increase the TCO gap between the two solutions. Ultimately, Diamanti can lower infrastructure TCO for Splunk solutions by a factor of two to three times.

Diamanti Cost	AWS Cost
$26/ GB	$70/ GB

Conclusion & Recommendations

Splunk needs to perform consistently and with speed to bring value to your organization. Falling victim to indexer latency, skipped searches, and slow searches can be detrimental to an organization's ability to gain accurate and real-time insights into their data. In order to deploy their Splunk Enterprise and Splunk Enterprise Security successfully, users need a solution that will scale with their organization needs.

Splunk on Diamanti using Kinney Group automation and optimization deploys faster and more reliably. Less human interaction reduces deployment risks. The Diamanti cluster offers top tier performance scaling to over double the recommended Splunk capacity for like hardware. Diamanti also proved to be more resilient to spikes and unpredictable system load. In the tests detailed above, the containers on Diamanti never reported using their maximum requested resources indicating that there is additional capacity to better handle high data velocity and volume on each indexer.

To conclude, results clearly show that Diamanti can outperform a similarly built AWS EKS cluster. Further, the Diamanti solution is simpler, deploys faster, and has a lower total cost of ownership.

Appendix I

SPLUNK COMPONENTS

Universal Forwarders

The Universal Forwarders (UF) deployed are configured with the Gogen App for Splunk to generate simulated data for ingestion by the indexing tier. Each forwarder is configured to generate approximately 250 GB/day. In order to increase the daily ingest rate, additional forwarders are created by scaling the deployment.

Universal Forwarders are configured with two (2) ingestion pipelines and unlimited maximum data throughout. While a typical deployment would utilize limits to minimize the impact of the forwarder on the host, because these are containers dedicated to generating and pushing data, those limits have been removed.

Indexers

Indexers are configured with a single ingestion pipeline, a replication factor of three (3), and a search factor of two (2) in a single site cluster to reflect Splunk recommended configurations. Sample data is being sent to the main index and replicated across the cluster. The required TA's are installed, as well as the Splunk_TA_ForIndexers generated by Splunk Enterprise Security.

Search Heads

Each environment includes a single search head with the appropriate TA's installed. We used the Splunk-Ansible playbook "configure_ess.yml" to install and set-up the Enterprise Security Suite. Then a set of correlation searches and data model acceleration searches corresponding to the ingested sourcetypes are manually enabled. The search head is manually configured as a distributed monitoring console as well.

In every test performed, the containers are deployed with identical configurations for consistency. All instances are configured per best practices recommended by Splunk.

Appendix II

SOFTWARE USED

The follow software packages were used in this design. These packages represent a typical deployment for an enterprise customer.

Premium Security Software

- Splunk Enterprise Security (ES)
- Splunk ES Supporting software
- Splunk Machine Learning Toolkit
- Splunk ES Content Updates and add-ons

Operating System Add-ons

- Windows
- Linux/Unix

Open Source Security Tool Add-ons

- Bro (Network detection & response)
- Bind (DNS)
- OSSec (Endpoint)
- Nmap (Network scanner)

Vendor Security Add-ons

- Bluecoat (Proxy)
- Cisco (Firewall, Proxy, and VPN)
- Juniper (Firewall, Network, and VPN)
- McAfee (Endpoint)
- SourceFire (Intrusion Detection/Intrusion Prevention)
- Symantec (Endpoint)
- Websense (Proxy)
- Alcatel (Voice)
- Crowdstrike (Endpoint)
- Fortinet (Firewall and VPN)
- Palo Alto Networks (Firewall, Network, and VPN)
- Tipping Point (Host Intrusion Prevention)
- Trend Micro (Endpoint)
- RSA SecureID (Authentication)
- Sophos (EndPoint
- Oracle (Databases)

During this test, the search skip ratio stays close to zero and execution latency stays below 30 seconds the entire test. Unsurprisingly, indexing shows minimal queue fill at this data rate and number of indexers.

About Us

Diamanti delivers purpose-built infrastructure for modern applications by supercharging Kubernetes with rapid Kubernetes deployments and transformational application performance.

Diamanti's fully-integrated Kubernetes platform gives platform architects, IT operations, and application owners the performance, simplicity, efficiency, and enterprise features necessary for cloud-native applications to go-to-market rapidly. Diamanti delivers the industry's first purpose-built, fully integrated Kubernetes platform, spanning on-premises and public cloud environments. The Diamanti platform is compatible with Splunk Validated Architectures (SVAs).

Kinney Group is a cloud solutions integrator specializing in analytics, automation, and hybrid cloud solutions. We design, build, and integrate IT infrastructure solutions for some of the most demanding government agencies and commercial organizations. By leveraging next-generation technologies, adopting proven engineering practices, and agile development principles, we create custom solutions and world-class environments for data.

Kinney Group, Inc. (KGI) is an award-winning, certified Splunk Elite Partner. Our team has experience working with Splunk deployments of all sizes, at various stages of execution, and across a variety of use cases. We've helped Commercial and Public Sector organizations design, develop, and implement Splunk at scale. This work includes guiding organizations with the design of their on-premise infrastructure for supporting the Splunk Enterprise platform.

DIAMANTI

KINNEY GROUP