

SPLUNK VALIDATED ARCHITECTURES

November 2018

Table of Contents

Introduction	1
Document Structure	2
Reasons to Use Splunk Validated Architectures	2
Pillars of Splunk Validated Architectures	3
What to Expect from Splunk Validated Architectures	3
Roles and Responsibilities	4
Overview of the Splunk Validated Architectures Selection Process	4
Step 1a: Define Your Requirements for Indexing and Search	6
Step 2a: Choose a Topology for Indexing and Search	11
Step 1b: Define Your Requirements for Data Collection	22
Step 2b: Select Your Data Collection Components	26
Step 3: Apply Design Principles and Best Practices	38
Summary & Next Steps	45
Next Steps	45
Appendix	46
Appendix "A": SVA Pillars Explained	46
Appendix "B": Topology Components	47

Introduction

Splunk Validated Architectures (SVAs) are proven reference architectures for stable, efficient, and repeatable Splunk deployments. Many of Splunk's existing customers have experienced rapid adoption and expansion, leading to certain challenges as they attempt to scale. At the same time, new Splunk customers are increasingly looking for guidelines and certified architectures to ensure that their initial deployment is built on a solid foundation. SVAs have been developed to help our customers with these growing needs.

Whether you are a new or existing Splunk customer, SVAs will help you build an environment that is easier to maintain and simpler to troubleshoot. SVAs are designed to provide you with the best possible results while minimizing your total cost of ownership. Additionally, your entire Splunk foundation will be based on a repeatable architecture which will allow you to scale your deployment as your needs evolve over time.

SVAs offer topology options that consider a wide array of organizational requirements, so you can easily understand and find a topology that is right for your requirements. The Splunk Validated Architectures selection process will help you match your specific requirements to the topology that best meets your organization's needs. If you are new to Splunk, we recommend implementing a Validated Architecture for your initial deployment. If you are an existing customer, we recommend that you explore the option of aligning with a Validated Architecture topology. Unless you have unique requirements that make it necessary to build a custom architecture, it is very likely that a Validated Architecture will fulfill your requirements while remaining cost effective.

This whitepaper will provide you with an overview of SVAs. Within this whitepaper you will find the resources you need to go through the SVA selection process, including the requirements questionnaire, deployment topology diagrams, design principles, and general guidelines.

If you need assistance implementing a Splunk Validated Architecture, contact [Splunk Professional Services](https://www.splunk.com/en_us/support-and-services/splunk-services.html) (https://www.splunk.com/en_us/support-and-services/splunk-services.html).

Document Structure

SVAs are broken into three major content areas:

1. Indexing and Search Topologies
2. Data Collection Architecture components
3. Design Principles and Best Practices

Indexing and search covers the architecture tiers that provide the core indexing and search capabilities of a Splunk deployment. The data collection component section guides you in choosing the right data collection mechanism for your requirements.

Design Principles and Best Practices apply to your architecture as a whole and will help you make the correct choices when working out the details of your deployment.

Reasons to Use Splunk Validated Architectures

Implementing a Validated Architecture will empower you to design and deploy Splunk more confidently. SVAs will help you solve some of the most common challenges that organizations face, including:

Performance

- Organizations want to see improvements in performance and stability.

Complexity

- Organizations sometimes run into the pitfalls of custom-built deployments, especially when they have grown too rapidly or organically. In such cases, unnecessary complexity may have been introduced into the environment. This complexity can become a serious barrier when attempting to scale.

Efficiency

- To derive the maximum benefits from the Splunk deployment, organizations must improve the efficiency of operations and accelerate time to value.

Cost

- Organizations are seeking ways to reduce total cost of ownership (TCO), while fulfilling all of their requirements.

Agility

- Organizations will need to adapt to change as they scale and grow.

Maintenance

- Optimization of the environment is often necessary in order to reduce maintenance efforts.

Scalability

- Organizations must have the ability to scale efficiently and seamlessly.

Verification

- Stakeholders within the organization want the assurance that their Splunk deployment is built on best practices.

Pillars of Splunk Validated Architectures

Splunk Validated Architectures are built on the following foundational pillars. For more information on these design pillars, refer to Appendix "A" below.

AVAILABILITY	PERFORMANCE	SCALABILITY	SECURITY	MANAGEABILITY
The system is continuously operational and able to recover from planned and unplanned outages or disruptions.	The system can maintain an optimal level of service under varying usage patterns.	The system is designed to scale on all tiers, allowing you to handle increased workloads effectively .	The system is designed to protect data, configurations, and assets while continuing to deliver value.	The system is centrally operable and manageable across all tiers .

These pillars are in direct support of the **Platform Management & Support** Service in the Splunk Center Of Excellence model.

What to Expect from Splunk Validated Architectures

Please note that SVAs do not include deployment technologies or deployment sizing. The reasoning for this is as follows:

- Deployment technologies, such as operating systems and server hardware, are considered implementation choices in the context of SVAs. Different customers will have different choices, so a generalization is not easily possible.
- Deployment sizing requires an evaluation of data ingest volume, data types, search volumes, and search use cases, which tend to be very customer-specific and generally have no bearing on the fundamental deployment architecture itself. Existing sizing tools can help with this process once you have established your deployment architecture. [Splunk Storage Sizing](https://splunk-sizing.appspot.com/) (<https://splunk-sizing.appspot.com/>) is one of the available tools.

SVAs <u>will</u> provide:	SVAs do <u>not</u> provide:
<ul style="list-style-type: none"> ✓ Clustered and non-clustered deployment options. ✓ Diagrams of the reference architecture. ✓ Guidelines to help you select the architecture that is right for you ✓ Tier-specific recommendations. ✓ Best practices for building out your Splunk deployment 	<ul style="list-style-type: none"> ✗ Implementation choices (OS, baremetal vs. virtual vs. Cloud etc.). ✗ Deployment sizing. ✗ A prescriptive approval of your architecture. Note: SVAs provide recommendations and guidelines, so you can ultimately make the right decision for your organization. ✗ A topology suggestion for every possible deployment scenario. In some cases, unique factors may require a custom architecture to be developed. Splunk experts are available to help with any custom solutions you need. If you are an existing customer, reach out to your Splunk Account Team. If you are new to Splunk, you can reach us here (https://www.splunk.com/en_us/talk-to-sales.html).

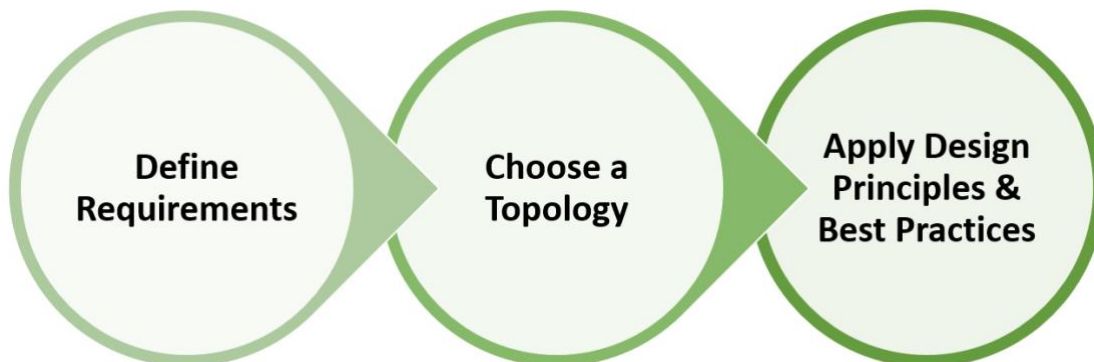
Roles and Responsibilities

Splunk Validated Architectures are highly relevant to the concerns of decision makers and administrators. Enterprise architects, consultants, Splunk administrators, and managed service providers should all be involved in the SVA selection process. You will find a description of each of these roles below:

Role	Description
Enterprise Architects	Responsible for architecting Splunk deployments to meet enterprise needs.
Consultants	Responsible for providing services for Splunk architecture, design, and implementation.
Splunk Engineers	Responsible for managing the Splunk lifecycle.
Managed Service Providers	Entities that deploy and run Splunk as a service for customers.

Overview of the Splunk Validated Architectures Selection Process

The Splunk Validated Architectures selection process will help you identify the simplest and most streamlined architecture that meets all of your organization's needs.



Steps in the Selection Process	Goals	Considerations
Step 1: Define Requirements for: a) Indexing and Search b) Data Collection Mechanism(s)	<i>Define requirements.</i>	<ul style="list-style-type: none"> Decision-makers, stakeholders, and admins should collaborate to identify and define your organization's requirements. If you already have a deployment in place, you can evaluate your current architecture to see what it would take to move to a validated model. <p><i>For a questionnaire that will help you define your requirements, refer to Step 1 below.</i></p>
Step 2: Choose a Topology for: a) Indexing and Search b) each data collection mechanism	<i>Choose a topology that meets identified requirements.</i>	<ul style="list-style-type: none"> You'll choose a topology that best meets your requirements. Keep things simple and in accordance with the SVA, so you can appreciate the easier path to scalability. <p><i>For diagrams and descriptions of topology options, refer to Step 2 below.</i></p>
Step 3: Apply Design Principles and Best Practices	<i>Prioritize your design principles and review tier-specific implementation best practices.</i>	<ul style="list-style-type: none"> Each design principle reinforces one or more of the pillars of Splunk validated architectures. You'll prioritize design principles in accordance with the needs of your organization. Tier-specific recommendations will guide your topology implementation. <p><i>For a breakdown of design principles, refer to Step 3 below.</i></p>

Step 1a: Define Your Requirements for Indexing and Search

To select the appropriate deployment topology, you will need to do a deep dive into your requirements. Once you have defined your requirements you will be able to choose the simplest, most cost-effective way to deploy Splunk. Below you will find a questionnaire that will help you define key requirements areas for the indexing and search tiers of your deployment.

The requirements questionnaire focuses on areas that will have a direct impact on your deployment topology. Therefore, we highly recommend that you record your answers to the questions below before choosing a topology in the next step.

Things to Keep Under Consideration

Review your use cases

As you define your requirements, you should think about the intended use cases for your Splunk infrastructure. For example, the topology for a departmental DevOps use case is often simpler than a mission-critical use case (though not in all cases).

You should fully consider use cases involving:

- Search
- Availability
- Compliance requirements (this is especially important if you need to have 100% data fidelity and availability at all times)
- Other use case scenarios specific to your organization

Depending on your use case scenarios, your deployment may need to provide additional architectural characteristics.

Think about future growth

You will need to think about your immediate needs in order to define your requirements. However, you should also consider future growth and scalability. Scaling your deployment may require expenditures, additional staffing, or other resources you may want to start planning for today.

Topology Categories

The following is a key to SVA topology categories. These categories are used in the questionnaire below. You will also find references to these categories in the next steps of the SVA selection process.

Indexing Tier Categories

Category Code	Explanation
S	Category "S" indicates the indexer of a single-server Splunk deployment
D	Category "D" indicates the need for a distributed indexer tier with at least 2 indexers
C	Category "C" indicates the need for a clustered indexer tier (data replication is required)
M	Category "M" indicates the need for a clustered indexer tier with multiple sites

Search Tier Categories

Category Code	Explanation
1	Category "1" indicates a single search head may meet the requirements
2	Category "2" indicates multiple search heads are required to meet requirement
3	Category "3" indicates a search head cluster is required to meet requirement
4	Category "4" indicates a search head cluster that spans multiple sites (a "stretched" SHC) is required to meet requirement
+10	Category "+10" indicates a dedicated search head (cluster) is required to support Enterprise Security App. Add 10 to the search tier topology category and carefully read the description for the topology for specific requirements for this app.

Questionnaire 1: Defining Your Requirements for Index and Search Tiers

♦ See the key above for an explanation of topology category codes. If you answer "yes" to multiple questions, use the topology category code for the highest numbered question.

#	Question	Considerations	Impact on Topology	Indexer Tier Topology Category ♦	Search Tier Topology Category ♦
1	Is your expected daily data ingest less than ~300GB/day?	Consider short-term growth in the daily ingest (~6-12 month)	Candidate for a single server deployment, depending on answers to availability-related questions	S	1
2	Do you require high availability for data collection/indexing?	If you are not planning on using Splunk for monitoring use cases that require continuous data ingest, a temporary interruption of the inbound data flow may be acceptable; assuming no log data is lost.	Requires distributed deployment to support continuous ingest	D	1
3	Assuming an available Search Head to run a search: Does your data need to be completely searchable at all times, i.e. you cannot afford any impact to search result completeness?	If your use case is calculating performance metrics and general usage monitoring using aggregate functions, for example, a single indexer outage may not materially affect	Requires clustered indexers with a replication factor of at least two (2). Note: While a replication factor of 2 provides minimal protection against a single indexer node failure, the	C	1

#	Question	Considerations	Impact on Topology	Indexer Tier Topology Category ♦	Search Tier Topology Category ♦
		the calculation of statistics over a large number of events. If your use case is security auditing and threat detection, blind spots in search results are very likely undesirable	recommended (and default) replication factor is 3.		
4	Do you operate multiple data centers and require automatic recovery of your Splunk environment in case of a data center outage?	Disaster recovery requirements may dictate continuous operation out of two facilities (active/active) or prescribe RTO/RPO goals for manual disaster recovery	Continuous operation will require multi-site indexer clustering and at least two active search heads to ensure failover at both the data ingest/indexing tier as well as the search tier.	M	2
5	Assuming continuous, lossless data ingest, do you require HA for the user-facing search tier?	If Splunk is being used for continuous, near-time monitoring, interruptions in the search tier are likely not tolerable. This may or may not be true for other use cases.	Requires redundant search heads, potentially search head clustering	D/C/M	3
6	Do you need to support a large number of concurrent users and/or a significant scheduled search workload?	Requirements for more than ~50 concurrent users/searches typically require horizontal scaling of the search tier	May require a topology that uses a search head cluster in the search tier	D/C/M	3
7	In a multi-data center environment, do you require user artifacts (searches, dashboards and other knowledge objects) to be synchronized between sites?	This will decide whether users will have a current and consistent experience in case of a site outage.	Requires a "stretched" search head cluster across sites with appropriate configuration. Important: While a stretched SHC can improve search availability for users during a complete site failure, it cannot be guaranteed that all	M	4

#	Question	Considerations	Impact on Topology	Indexer Tier Topology Category ♦	Search Tier Topology Category ♦
			artifacts are replicated across both sites at all times. This may affect specific applications that rely on consistent and current artifacts, like the Splunk App for Enterprise Security. Search Head Clustering alone cannot provide a complete DR solution. Other benefits for SHC still apply.		
8	Are you intending to deploy the Splunk App for Enterprise Security (ES)?	Please ensure you <u>read and understand</u> the specific limitations the Splunk App for Enterprise Security is subject to as documented with each topology.	ES requires a dedicated Search Head environment (either standalone or clustered).	D/C/M	+10
9	Do you have a geographically distributed environment that is subject to data custody regulations?	Some countries' regulations do not allow data generated within the country to leave systems in that country.	Such regulations prevent deployment of a central Splunk indexing tier and require a custom architecture to be developed by collaboration between Splunk/Partner and the customer that considers the details of such a deployment in depth. In other words, there is no SVA to meet this requirement.	Custom	Custom
10	Do you have highly restrictive security policies that prevent co-location of specific log data sources on shared servers/indexers?	Highly sensitive log data may not be allowed to be co-located with lower-risk datasets on the same physical system/within the same network zone based on corporate policies.	Multiple, independent indexing environments are needed, potentially with a shared, hybrid search tier. This is beyond the scope of SVAs and requires custom architecture development.	Custom	Custom

How to Determine Your Topology Category Code

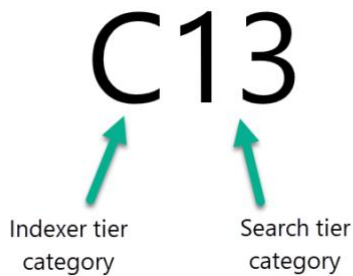
Based on your answers to the requirements questionnaire above, you will end up with a combined topology category indicator that will allow you to identify the best topology for your needs. Instructions and examples are provided below.

Instructions

1. Write down the questions to which you answered "yes".
2. If you answered "yes" to multiple questions, follow the topology recommendation for the highest numbered question. If you see multiple topology options (for example, "D/C/M"), look at the previous questions to determine which option is best suited for you.
3. Your topology category code will begin with the letter representing the the indexer tier (for example, "C" or "M"). This letter will be followed by the number representing the search tier (for example, "1" or "13").

Example #1

Let's say you answered "yes" to questions #3, #5 and #8. You will will end up with a topology category of "C13", indicating the need for a clustered indexing tier with two search head clusters.



Example #2

Now, let's say you answered "yes" only to question #1. You will end up with a topology category of "S1", indicating a single-server Splunk deployment as your ideal topology.



Step 2a: Choose a Topology for Indexing and Search

Topologies are generally split into non-clustered and clustered deployments. Non-clustered deployments require the least amount of distinct components and have excellent scalability characteristics. Keep in mind that even though non-clustered deployments come with reduced availability and disaster recovery features, this deployment option may still be a good choice for your organization.

Remember: The primary goal of the SVA selection process is to allow you to build what you need without introducing unnecessary components.

Note

While you may choose to implement a topology that provides additional benefits beyond your immediate needs, keep in mind that this will likely result in unnecessary costs. Moreover, the introduction of additional complexity is often counter-productive to operational efficiency.

Important Note about topology diagrams

The icons in the topology diagrams represent **functional Splunk roles** and do not imply dedicated infrastructure to run them. Please see the Appendix for guidance as to which Splunk roles can be colocated on the same infrastructure/server.

Using Your Topology Category Code

Before selecting a topology option, it is highly recommended that you complete the requirements questionnaire to determine your topology category code. If you have not yet done this, please go back and complete the previous step above. Once you have your topology category code you will be able to identify the deployment option that is the best fit for your stated requirements.

Non-clustered deployment options

Below you will find the following topology options:

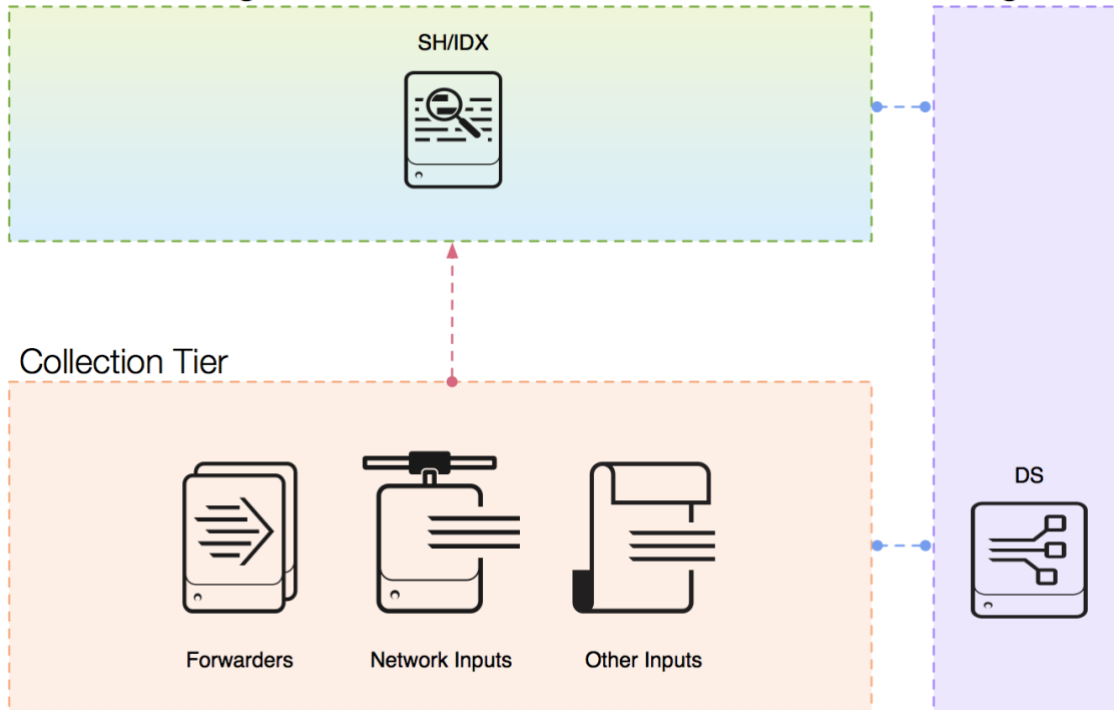
Type of Deployment	Topology Category Code(s)
Single Server Deployment	S1
Distributed Non-Clustered Deployment	D1 / D11

For an explanation of topology components, refer to Appendix "B" below.

Single Server Deployment (S1)

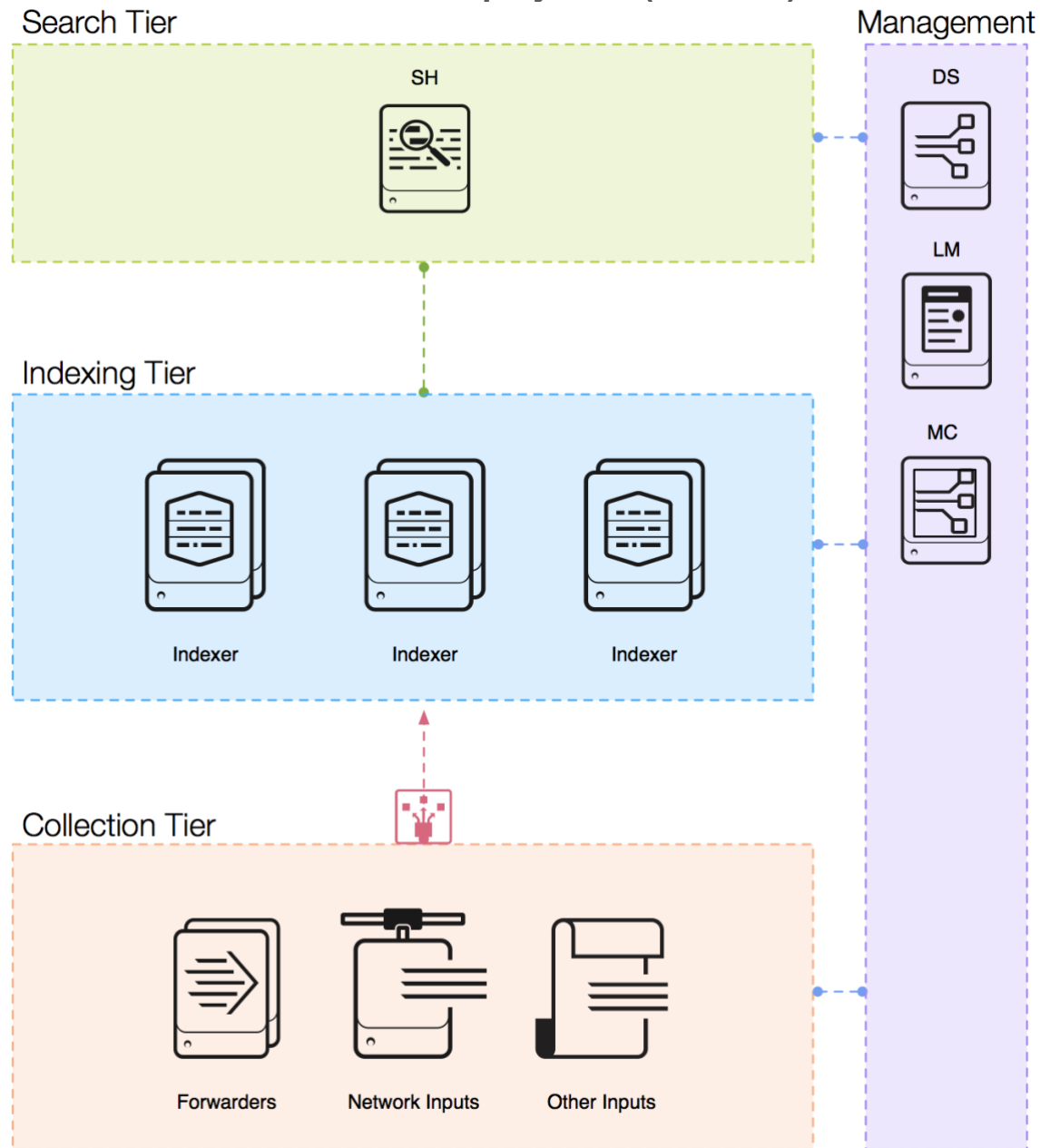
Search/Indexing Tier

Management



Description of the Single Server Deployment (S1)	Limitations
<p>This deployment topology provides you with a very cost-effective solution if your environment meets all of the following criteria: a) you do not have any requirements to provide high-availability or automatic disaster recovery for your Splunk deployment, b) your daily data ingest is under 300GB/day, and c) you have a small number of users with non-critical search use cases.</p> <p>This topology is typically used for smaller, non business-critical use-cases (often departmental in nature). Appropriate use cases include data onboarding test environments, small DevOps use cases, application test and integration environments, and similar scenarios.</p> <p>The primary benefits of this topology include easy manageability, good search performance for smaller data volumes, and a fixed TCO.</p>	<ul style="list-style-type: none"> • No High Availability for Search/Indexing • Scalability limited by hardware capacity (straightforward migration path to a distributed deployment)

Distributed Non-Clustered Deployment (D1 / D11)



Description of the Distributed Non-Clustered Deployment (D1 / D11)	Limitations
<p>You need to move to a distributed topology in either of the following situations: a) your daily data volume to be sent to Splunk exceeds the capacity of a single-server deployment, or b) you want/need to provide highly available data ingest. Deploying multiple, independent indexers will allow you to scale your indexing capacity linearly and implicitly increase the availability for data ingest.</p> <p>The TCO will increase in a predictable and linear fashion as you add indexer nodes. The recommended introduction of the Monitoring Console (MC) component allows you to monitor the</p>	<ul style="list-style-type: none"> • No High Availability for Search Tier • Limited High Availability for indexing tier, node failure may cause incomplete search results for historic searches

Description of the Distributed Non-Clustered Deployment (D1 / D11)	Limitations
<p>health and capacity of your distributed deployment. Additionally, the MC provides a centralized alerting system so you will be notified of unhealthy conditions in your deployment.</p> <p>The search head(s) will need to be configured manually with the list of available search peers every time new indexers are added. Note for ES Customers: If your category code is D1 (i.e. you intend to deploy the Splunk App for Enterprise Security), a single dedicated search head is required to deploy the app (this is not pictured in the topology diagram).</p> <p>The collection tier needs to be configured with the list of target indexers (via a deployment server) every time new indexers are added.</p> <p>This deployment topology can scale linearly to over 1000 indexer nodes and can thus support extremely high data ingest and search volumes.</p> <p>Search performance can be maintained across large datasets through parallel search execution across many indexers (map/reduce).</p> <p>While not specifically broken out as a separate topology, a search head cluster can be used to increase search capacity on the search tier (refer to the search tier in topology C3/C13).</p>	

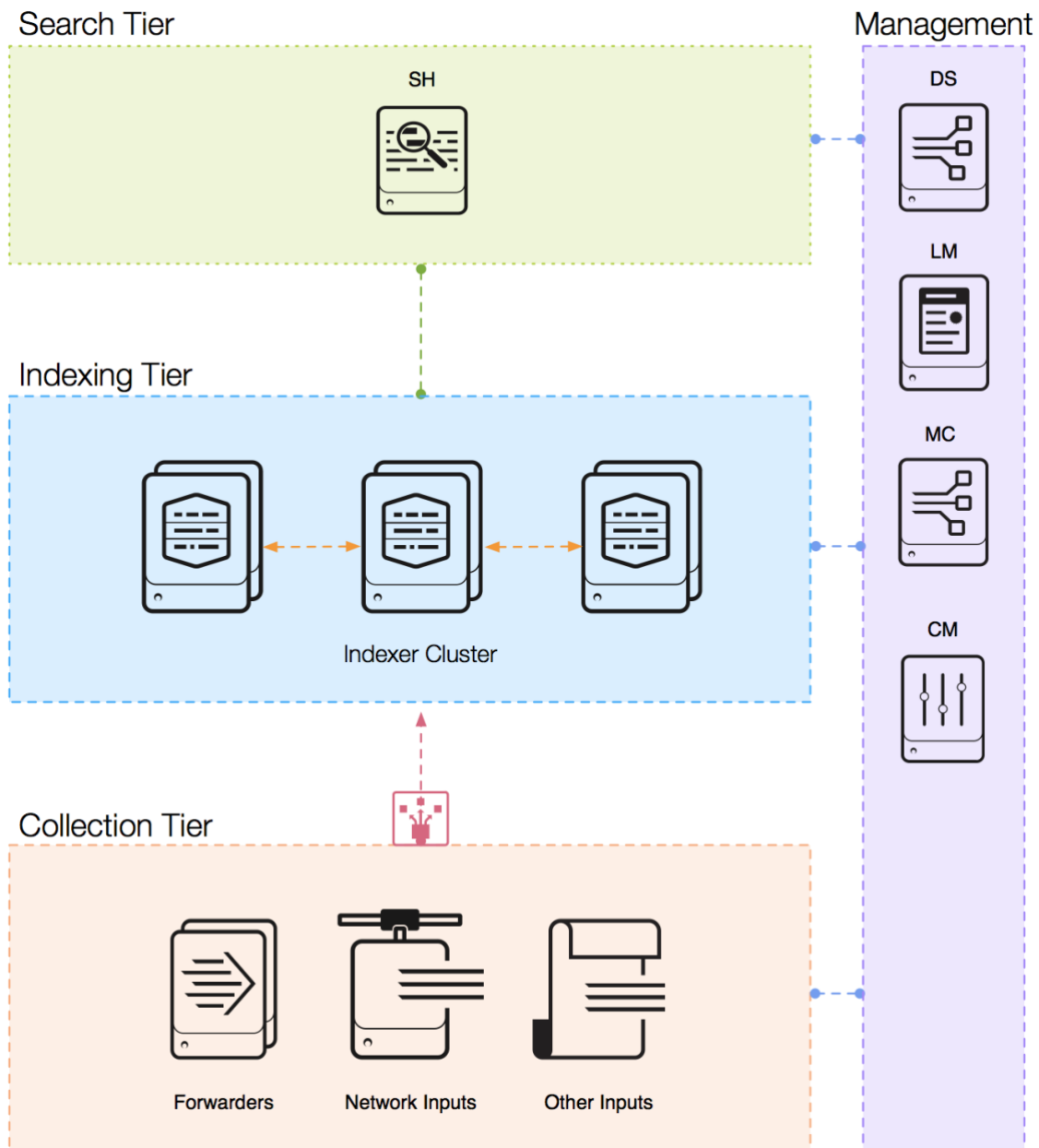
Clustered deployment options

Below you will find the following topology options:

Type of Deployment	Topology Category Code(s)
Distributed Clustered Deployment - Single Site	C1 / C11
Distributed Clustered Deployment + SHC - Single Site	C3 / C13
Distributed Clustered Deployment - Multi-Site	M2 / M12
Distributed Clustered Deployment + SHC - Multi-Site	M3 / M13
Distributed Clustered Deployment + SHC - Multi-Site	M4 / M14

For an explanation of topology components, see Appendix "B" below.

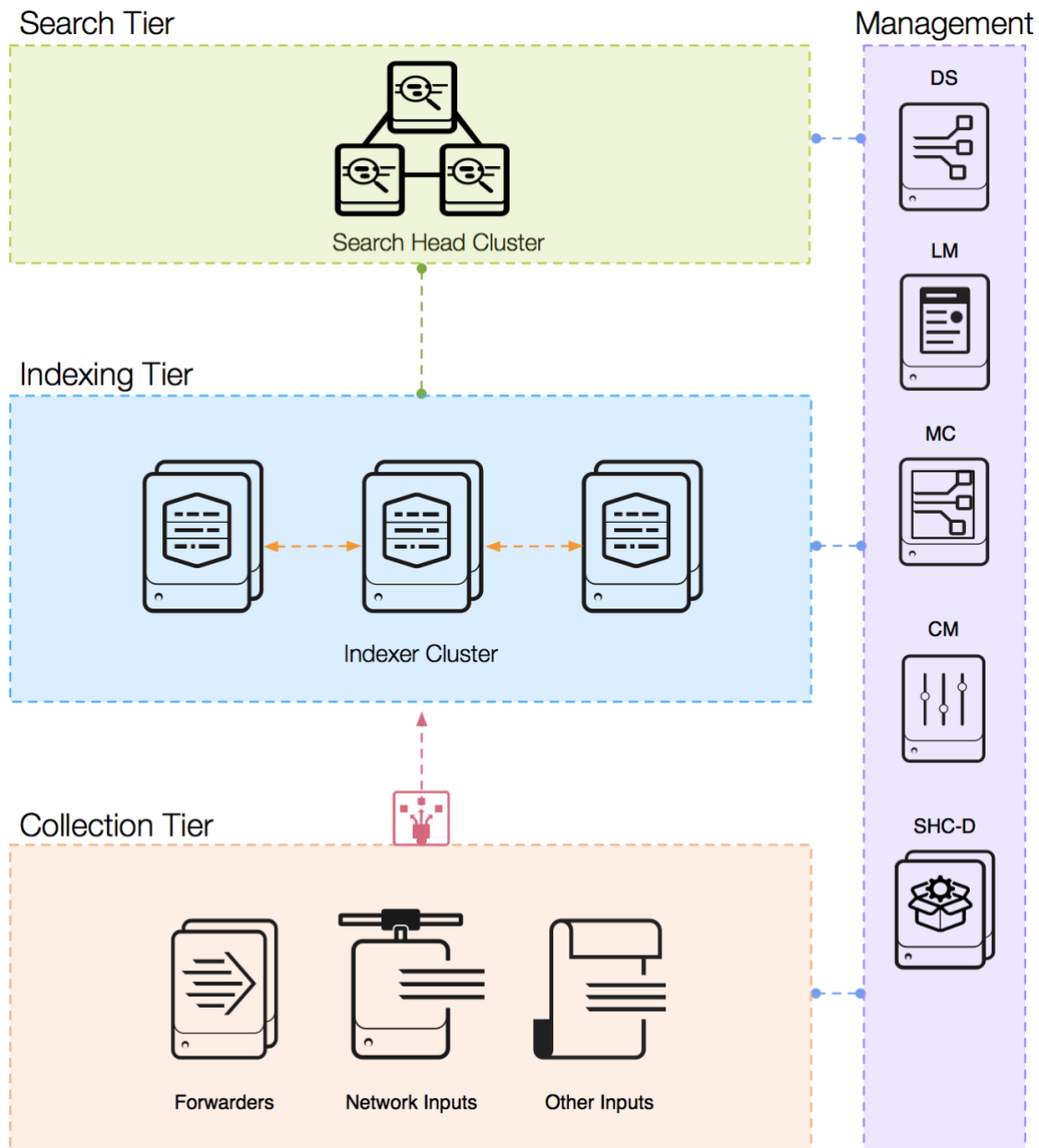
Distributed Clustered Deployment - Single Site (C1 / C11)



Description of Distributed Clustered Deployment - Single Site (C1 / C11)	Limitations
<p>This topology introduces indexer clustering in conjunction with an appropriately configured data replication policy. This provides high-availability of data in case of indexer peer node failure. However, you should be aware that this applies only to the indexing tier and does not protect against search head failure.</p> <p>Note for ES customers: If your category code is C11 (i.e. you intend to deploy the Splunk App for Enterprise Security), a single dedicated</p>	<ul style="list-style-type: none"> • No High Availability for Search Tier • Total number of unique buckets in indexer cluster limited to 5MM (V6.6+), 15MM total buckets

Description of Distributed Clustered Deployment - Single Site (C1 / C11)	Limitations
<p>search head is required to deploy the app (this is not pictured in the topology diagram).</p> <p>This topology requires an additional Splunk component named the Cluster Master (CM). The CM is responsible for coordination and enforcement of the configured data replication policy. The CM also serves as the authoritative source for available cluster peers (indexers). Search Head configuration is simplified by configuring the CM instead of individual search peers.</p> <p>You have the option of configuring the forwarding tier to discover available indexers via the CM. This simplifies the management of the forwarding tier.</p> <p>Be aware that data is replicated within the cluster in a non-deterministic way. You will not have control over where requested copies of each event are stored. Additionally, while scalability is linear, there are limitations with respect to total cluster size (~50PB of searchable data under ideal conditions).</p> <p>We recommend deployment of the Monitoring Console (MC) to monitor the health of your Splunk environment.</p>	<ul style="list-style-type: none">• No automatic DR capability in case of data center outage

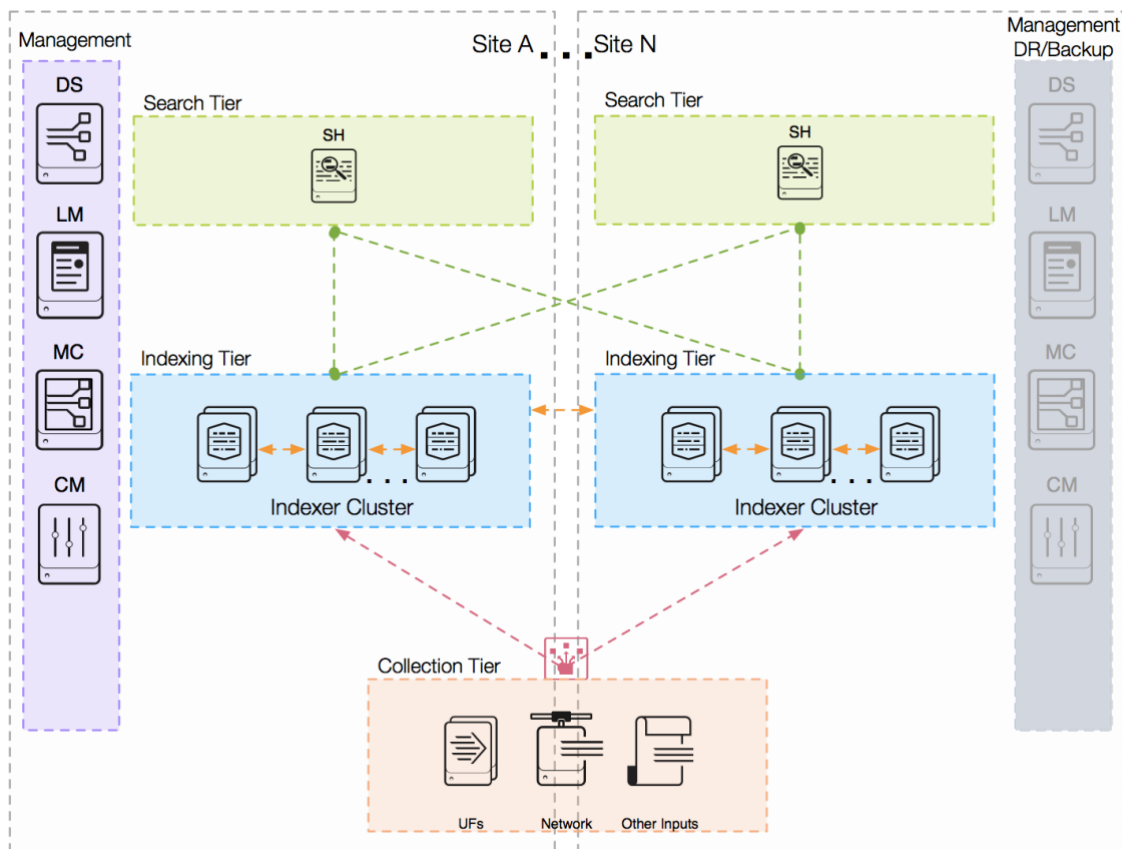
Distributed Clustered Deployment + SHC - Single Site (C3 / C13)



Description of Distributed Clustered Deployment + SHC - Single Site (C3 / C13)	Limitations
<p>This topology adds horizontal scalability and removes the single point of failure from the search tier. A minimum of three search heads are required to implement a SHC.</p> <p>To manage the SHC configuration, an additional Splunk component called the Search Head Cluster Deployer is required for each SHC. This component is necessary in order to deploy changes to configuration files in the cluster. The Search Head Cluster Deployer has no HA requirements (no runtime role).</p>	<ul style="list-style-type: none"> • No DR capability in case of data center outage • ES requires dedicated SH/SHC • Managing an ES deployment on SHC supported, but

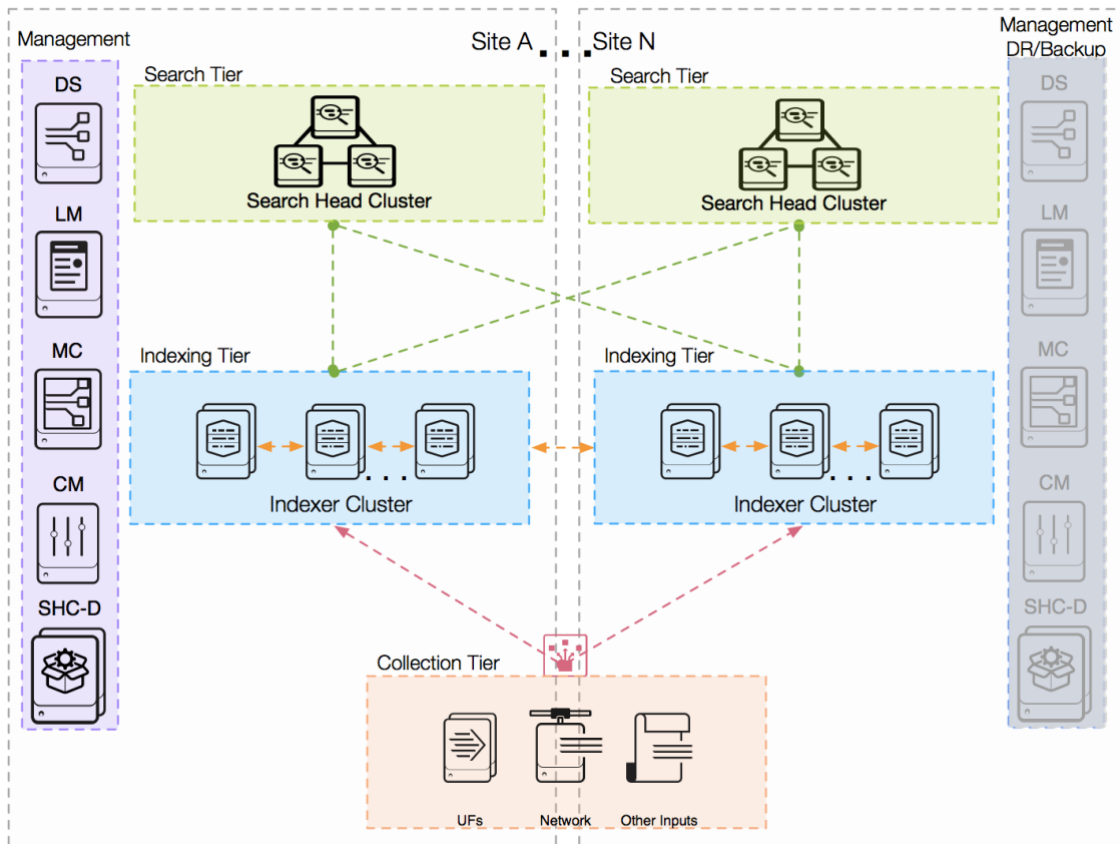
Description of Distributed Clustered Deployment + SHC - Single Site (C3 / C13)	Limitations
<p>The SHC provides the mechanism to increase available search capacity beyond what a single search head can provide. Additionally, the SHC allows for scheduled search workload distribution across the cluster. The SHC also provides optimal user failover in case of a search head failure.</p> <p>A network load-balancer that supports sticky sessions is required in front of the SHC members to ensure proper load balancing of users across the cluster.</p> <p>Note for ES customers: If your category code is C13 (i.e. you intend to deploy the Splunk App for Enterprise Security), a dedicated search head cluster is required to deploy the app (this is not pictured in the topology diagram). The search tier can contain clustered and non-clustered Search Heads depending on your capacity and organizational needs (this is also not pictured in the topology diagram).</p>	<p>challenging (Involve PS)</p> <ul style="list-style-type: none"> SHC cannot have more than 100 nodes

Distributed Clustered Deployment - Multi-Site (M2 / M12)



Description of Distributed Clustered Deployment - Multi-Site (M2 / M12)	Limitations
<p>To provide near-automatic disaster recovery in case of a catastrophic event (like a data center outage), multi-site clustering is the deployment architecture of choice. A healthy multi-site cluster requires acceptable inter-site network latency as specified in the Splunk documentation.</p> <p>This topology allows you to deterministically replicate data to two or more groups of indexer cluster peers. You will be able to configure the site replication and search factor. This site-replication factor allows you to specify where replica copies are being sent to and ensures data is distributed across multiple locations.</p> <p>It is still managed by a single cluster master node, which has to be failed over to the DR site in case of a disaster.</p> <p>Multi-site clustering provides data redundancy across physically separated distributed locations, with the possibility for geographically separated distribution.</p> <p>Users can fail over to the DR site automatically to ensure availability. However, this topology does not provide a mechanism to automatically synchronize the search tier configuration and runtime artifacts across sites.</p> <p>Available search peer (indexer) capacity across sites can be utilized for search execution in an active/active model. When possible, site-affinity can be configured to ensure that users logged on to a specific site's search head will only search local indexers.</p> <p>Note for ES customers: If your category code is M12 (i.e. you intend to deploy the Splunk App for Enterprise Security), a single dedicated search head is required to deploy the app (this is not pictured in the topology diagram). For the ES search head, failover involves setting up a "shadow" search head in the failover site that is only activated and used in a DR situation. Please engage Splunk Professional Services to design and implement a site failover mechanism for your Enterprise Security deployment.</p>	<ul style="list-style-type: none"> • No sharing of available Search Head capacity and no search artifact replication across sites • Failure of Management functions need to be handled outside of Splunk in case of site failure • Cross-site latency for index replication must be within recommended limits

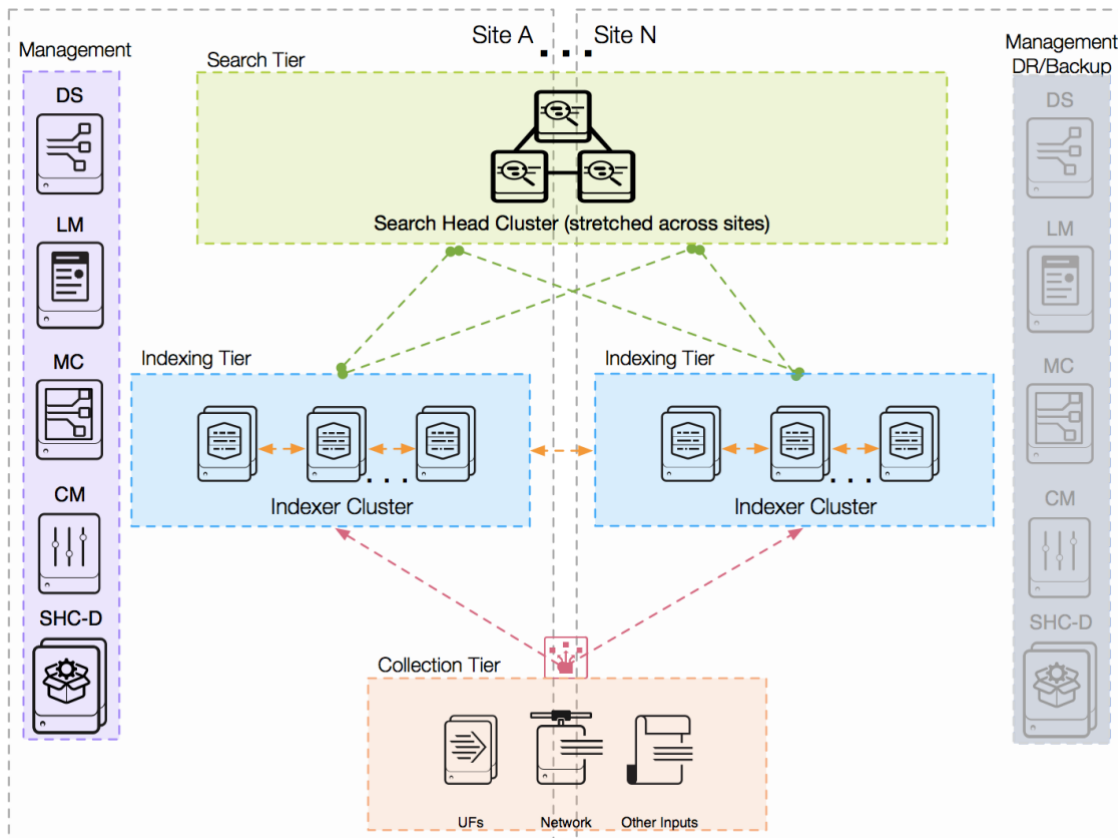
Distributed Clustered Deployment + SHC - Multi-Site (M3 / M13)



Description of Distributed Clustered Deployment + SHC - Multi-Site (M3 / M13)	Limitations
<p>This topology adds horizontal scalability and removes the single point of failure from the search tier in each site. A minimum of three search heads are required to implement a SHC (per site).</p> <p>To manage the SHC configuration, an additional Splunk component called the Search Head Cluster Deployer is required for each SHC. This component is necessary in order to deploy changes to configuration files in the cluster. The Search Head Cluster Deployer has no HA requirements (no runtime role).</p> <p>The SHC provides the following benefits: a) increased available search capacity beyond what a single search head can provide, b) scheduled search workload distribution across the cluster, and c) optimal user failover in case of a search head failure.</p> <p>A network load-balancer that supports sticky sessions is required in front of the SHC members in each site to ensure proper load balancing of users across the cluster.</p> <p>Note for ES customers: If your category code is M13 (i.e. you intend to deploy the Splunk App for Enterprise Security), a single dedicated search head cluster <i>contained within a site</i> is required to deploy the app (this is not explicitly pictured in the topology diagram). To be able to recover an ES SH environment from a site failure, 3rd party technology can be used to perform a failover of the search head instances, or a</p>	<ul style="list-style-type: none"> • No search artifact replication across sites, SHCs are standalone • Cross-site latency for index replication must be within documented limits • SHC cannot have more than 100 nodes

Description of Distributed Clustered Deployment + SHC - Multi-Site (M3 / M13)	Limitations
"warm standby" ES SH can be provisioned and kept in synch with the primary ES environment. It is strongly recommended to engage with Splunk Professional Services when deploying ES in a HA/DR environment.	

Distributed Clustered Deployment + SHC - Multi-Site (M4 / M14)



Description of Distributed Clustered Deployment + SHC - Multi-Site (M4 / M14)	Limitations
<p>This is the most complex validated architecture, designed for deployments that have strict requirements around high-availability and disaster recovery. We strongly recommend involving Splunk Professional Services for proper deployment. When properly deployed, this topology provides continuous operation of your Splunk infrastructure for data collection, indexing, and search.</p> <p>This topology involves implementation of a "stretched" search head cluster that spans one or more sites. This provides optimal failover for users in case of a search node or data center failure. Search artifacts and other runtime knowledge objects are replicated in the SHC. Careful configuration is required to ensure that replication will happen across sites, as the SHC itself is not site-aware (i.e. artifact replication is non-deterministic).</p>	<ul style="list-style-type: none"> Network latency across sites must be within documented limits Failover of the SHC may require manual steps if only a minority of cluster members survive

Description of Distributed Clustered Deployment + SHC - Multi-Site (M4 / M14)	Limitations
<p>Site-affinity can be configured to ensure the WAN link between sites is utilized only in cases when a search cannot be satisfied locally.</p> <p>A network load-balancer that supports sticky sessions is required in front of the SHC members to ensure proper load balancing of users across the cluster.</p> <p>Note for ES customers: If your category code is M14 (i.e. you intend to deploy the Splunk App for Enterprise Security), a single dedicated search head cluster <i>contained within a site</i> is required to deploy the app (this is not explicitly pictured in the topology diagram). ES requires a consistent set of runtime artifacts to be available and this cannot be guaranteed in a stretched SHC when a site outage occurs. To be able to recover an ES SH environment from a site failure, 3rd party technology can be used to perform a failover of the search head instances, or a "warm standby" ES SH can be provisioned and kept in synch with the primary ES environment. It is strongly recommended to engage with Splunk Professional Services when deploying ES in a HA/DR environment.</p>	

Step 1b: Define Your Requirements for Data Collection

The data collection tier is a core component of a Splunk deployment. It enables any device in your environment to forward data to the indexing tier for processing, thereby making it available for search in Splunk. The most important factor here is to ensure that forwarding and indexing happen in the most efficient and reliable way, as this is critical to the success and performance of your Splunk deployment.

Consider the following aspects for your data collection tier architecture:

- The origin of your data. Does it come from log files, syslog sources, network inputs, OS event logging facilities, applications, message bus or elsewhere?
- Requirements for data ingest latency and throughput
- Ideal event distribution across the indexers in your indexing tier
- Fault tolerance and automatic recovery (HA)
- Security and data sovereignty requirements

This section of SVAs focuses on the common data collection methods. This section also discusses architecture and best practices for each data collection method and calls out potential issues to consider when making your implementation choice.

Important Architectural Considerations and Why They Matter

Given the essential role of the data collection tier, it's important to understand the key considerations involved in designing the architecture.

While some of these considerations may or may not be relevant to you based on your requirements, the considerations in bold text in the table below describe fundamental items that are relevant for every environment.

Consideration	Why is this important?
Data is ingested properly (timestamps, line breaking, truncation)	The importance of ideal event distribution across indexers cannot be overstated. The indexing tier works most efficiently when all available indexers are equally utilized. This is true for both data ingest as well as search performance. A single indexer that handles significantly more data ingest compared to peers can negatively impact search response times. For indexers with limited local disk storage, uneven event distribution may also cause data to be prematurely aged out before meeting the configured data retention policy.
Data is optimally distributed across available indexers	If data is not ingested properly because event timestamps and line breaking are not correctly configured, searching this data will become very difficult. This is because event boundaries have to be enforced at search time. Incorrect or missing timestamp extraction configurations can cause unwanted implicit timestamp assignment. This will confuse your users and make getting value out of your data much more difficult than it needs to be.
All data reaches the indexing tier reliably and without loss	Any log data that is collected for the purpose of reliable analytics needs to be complete and valid, such that searches performed on the data provide valid and accurate results.
All data reaches the indexing tier with minimum latency	Delays in data ingest will increase the time between a potentially critical - event occurring and the ability to search for and react to it. Minimal ingest latency is often crucial for monitoring use cases that trigger alerts to staff or incur automated action.
Data is secured while in transit	If the data is either sensitive or has to be protected while being sent over non-trusted networks, encryption of data may be required to prevent unauthorized third-party interception. Generally, we recommend all connections between Splunk components to be SSL enabled.
Network resource use is minimized	The network resource impact of log data collection must be minimized so as not to impact other business critical network traffic. For leased-line networks, minimizing network utilization also contributes to a lower TCO of your deployment.
Authenticate/authorize data sources	To prevent rogue data sources from affecting your indexing environment, consider implementing connection authentication/authorization. This may be covered by using network controls, or by employing application-level mechanisms (e.g., SSL/TLS).

Because of its vital role for your deployment, the guidance in this document focuses on architectures that support ideal event distribution. When a Splunk environment does not provide expected search performance, it is in almost all cases either caused by not meeting minimum storage performance requirements and/or uneven event distribution that limits exploiting search parallelization.

Now that you understand the most critical architectural considerations, let's find out what specific data collection requirements you need to fulfill.

Questionnaire 2: Defining Your Requirements for Data Collection

Answering the following questions will give you a list of data collection components you need in your deployment. You can use the keys in the rightmost column to find more details about each component further down in the document.

#	Question	Considerations	Impact on Topology	Relevant Data Collection Components
1	Do you need to monitor local files or execute data collection scripts on endpoints?	This a core requirement for almost all Splunk deployment scenarios.	You will need to install the universal forwarder on your endpoints and manage its configuration centrally.	UF
2	Do you need to collect log data sent via syslog from devices that you cannot install software on (appliances, network switches, etc.)?	Syslog is a ubiquitous transport protocol often used by purpose-built devices that do not allow installation of custom software.	You will need a syslog server infrastructure that serves as the collection point.	SYSLOG HEC
3	Do you need to support collection of log data from applications that log to an API versus writing to local disks?	Writing to log files on endpoints requires providing disk space and management of these log files (rotation, deletion, etc.). Some customers want to get away from this model and log directly to Splunk using available logging libraries.	You will need to use the Splunk HTTP event collector (HEC) or another technology that serves as a log sink.	HEC
4	Do you need to collect data from a streaming event data provider?	Many enterprises have adopted an event hub model where a centralized streaming data platform (like AWS Kinesis or Kafka) serves as the message transport between log data producers and consumers.	You will need an integration between the streaming data provider and Splunk.	KAFKA KINESIS HEC
5	Do you have non-negotiable security policies that prevent log producers to establish TCP connections directly with the indexing tier?	Sometimes, network topologies consist of multiple network zones with restrictive firewall rules between them and it may not be possible to generically allow traffic on Splunk ports to flow between zones. Configuring and maintaining firewall rules for individual source/target IP addresses would be to cumbersome.	You will need an intermediary forwarding tier that allows traffic to flow between network zones.	IF
6	Do you need to collect log data using programmatic means, e.g., by	Splunk provides various modular inputs that allow execution of scripts against APIs for a wide variety of data	Your data collection tier will require one or more data collection nodes (DCN) implemented with	DCN

	calling REST APIs or querying databases?	ingestion use cases, including DBX for collecting data from relational databases.	a Splunk Heavy Forwarder.	
7	Do you need to route (a subset of) data to other systems besides — and in addition to — Splunk?	Some use cases require data that is indexed in Splunk to also be forwarded to another system. Often, the forwarded data consists of only a subset of the source data, or the data has to be modified before being forwarded.	Depending on the use case specifics, you may need an intermediary forwarding tier built with a Heavy Forwarder to support event-based routing and filtering. Alternatively, you can forward data post-indexing by using the cefout command contained in the Splunk App for CEF.	HF
8	Do you have remote sites with network bandwidth constraints and require significant filtering of data before it is being sent over the network?	Filtering data before transmitting requires a parsing (heavy) forwarder. The outbound network bandwidth used by a HWF is about 5x that of a UF, so filtering only makes sense if a significant number of events are filtered out (rule of thumb: >50% of source data). Ideally, you should adjust your logging granularity to achieve the needed reduction in log volume.	If you cannot reduce your log volume at the source, you will need an intermediary HF at your remote site that parses source data and filters out events based on configuration.	IF HF
9	Do you need to mask/obfuscate sensitive data before it is sent over a public network for indexing?	Sometimes, securing forwarder traffic with SSL is not enough to protect sensitive data in transit over public networks and individual parts of events must be masked before transmission (SSNs, CC data, etc.). Ideally, such data masking will be done in the application that produces the log data.	If you cannot mask data in the producing application, you will need an intermediary HF at your site that parses source data and applies the required masking rules based on configuration before the data is sent to indexers.	IF HF
10	Do you need to capture metrics using statsd or collectd?	Statsd and collectd are ubiquitous technologies in use to gather metrics from host systems and applications.	Splunk supports specific index types and collection methods to feed those indices using either UF, HF or HEC.	METRICS
11	Do you require any of your data collection components to be highly available?	Typically not applicable for endpoints, availability may be a concern for other data collection components, like intermediary forwarders or data collection nodes.	Thought needs to be given to how outages will affect the availability of each component, and how to address it.	HA

Step 2b: Select Your Data Collection Components

After you complete the questionnaire, you will have a list of the required data collection components to meet the requirements for your deployment. This section discusses each data collection architecture component in more detail. Before we do that, let's briefly provide some general guidance.

General Forwarding Architecture Guidance

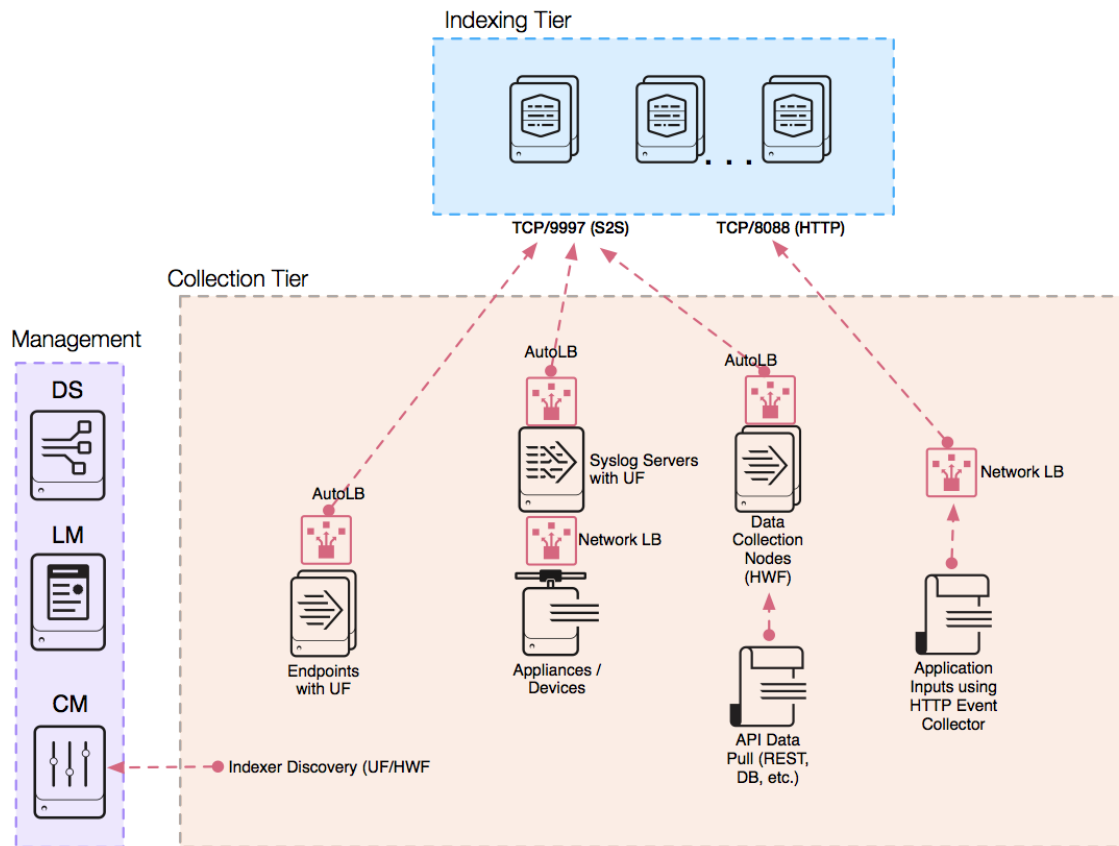
Ideally, the data collection tier is as "flat" as possible. This means that data sources are collected locally by a universal forwarder and forwarded directly to the indexing tier. This is a best practice because it ensures minimal data ingest latency (time to search) and enables proper event distribution across available indexers. Following this best practice leads to ease of management and operational simplicity. We often see customers deploy an intermediary forwarding tier. Generally speaking, avoid this unless requirements cannot be met otherwise. Due to the potential impact of intermediary forwarders, this document contains a separate section on this topic with more details.

There are endpoints that do not allow installation of the universal forwarder (in other words, network devices, appliances) and log using the syslog protocol. A separate best practice architecture to collect such data sources is outlined in the section titled Syslog Data Collection.

For data sources that have to be collected using programmatic means (APIs, database access), deploying a data collection node (DCN) based on a full Splunk enterprise install is recommended. This is also known as a heavy forwarder. It is not recommended that you run these kinds of inputs on the search head tier in anything other than a development environment.

The following diagram shows a general data collection architecture that reflects this guidance.

Data Collection Topology Overview



The diagram above shows the Deployment Server (DS) in the management tier, which is used to manage the configurations on data collection components. Also, the License Master (LM) is shown here since data collection nodes require access to the LM to enable Splunk Enterprise features. The cluster master (CM), if available, can be used by forwarders for indexer discovery, removing the need to manage available indexers in the forwarder output configuration.

In the above diagram, AutoLB represents the Splunk built-in auto-load balancing mechanism. This mechanism is used to ensure proper event distribution for data sent using the Splunk proprietary S2S protocol (default port 9997). Note: Using an external network load-balancer for S2S traffic is currently not supported and not recommended.

To load-balance traffic from data sources that communicate with an industry-standard protocol (like HTTP or syslog), a network load balancer is used to ensure even load and event distribution across indexers in the indexing tier.

(UF) Universal Forwarder

The universal forwarder (UF) is the best choice for a large set of data collection requirements from systems in your environment. It is a purpose-built data collection mechanism with very minimal resource requirements. The UF should be the default choice for collecting and forwarding log data. The UF provides:

- Checkpoint/restart function for lossless data collection.
- Efficient protocol that minimizes network bandwidth utilization.

- Throttling capabilities.
- Built-in, load-balancing across available indexers.
- Optional network encryption using SSL/TLS.
- Data compression (use only without SSL/TLS).
- Multiple input methods (files, Windows Event logs, network inputs, scripted inputs).
- Limited event filtering capabilities (Windows event logs only).
- Parallel ingestion pipeline support to increase throughput/reduce latency.

With few exceptions for well-structured data (json, csv, tsv), the UF does not parse log sources into events, so it cannot perform any action that requires understanding of the format of the logs. It also ships with a stripped down version of Python, which makes it incompatible with any modular input apps that require a full Splunk stack to function.

It is normal for a large number of UFs (100s to 10,000s) to be deployed on endpoints and servers in a Splunk environment and centrally managed, either with a Splunk deployment server, or a third-party configuration management tool (like e.g. Puppet or Chef).

(HF) Heavy Forwarder

The heavyweight forwarder (HWF) is a full Splunk Enterprise deployment configured to act as a forwarder with indexing disabled. A HWF generally performs no other Splunk roles. The key difference between a UF and a HWF is that the HWF contains the full parsing pipeline, performing the identical functions an indexer performs, without actually writing and indexing events on disk. This enables the HWF to understand and act on individual events, for example to mask data or to perform filtering and routing based on event data. Since it is a full Splunk Enterprise install, it can host modular inputs that require a full Python stack to function properly for data collection or serve as an endpoint for the Splunk HTTP event collector (HEC). The HWF performs the following functions:

- Parses data into events.
- Filters and routes based on individual event data.
- Has a larger resource footprint than the UF.
- Has a larger network bandwidth footprint than the UF (~5x).
- GUI for management.

In general, HWFs are not installed on endpoints for the purpose of data collection. Instead, they are used on standalone systems to implement data collection nodes (DCN) or intermediary forwarding tiers. **Use a HWF only when requirements to collect data from other systems cannot be met with a UF.**

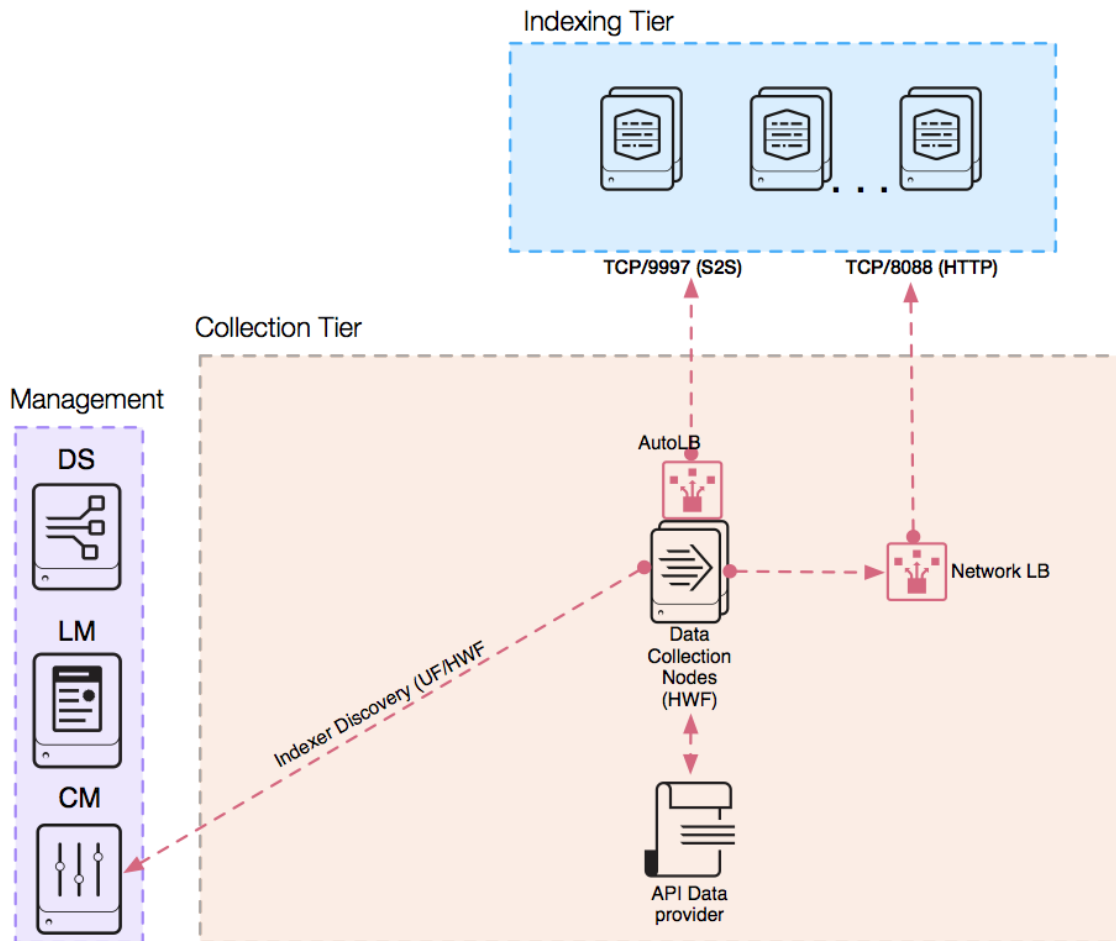
Examples of such requirements include:

- Reading data from RDBMS for the purposes of ingesting it into Splunk (database inputs).
- Collecting data from systems that are reachable via an API (cloud services, VMWare monitoring, proprietary systems, etc.).
- Providing a dedicated tier to host the HTTP event collector service .
- Implementing an intermediary forwarding tier that requires a parsing forwarder for routing/filtering/masking.

(DCN) Heavy Forwarder as Data Collection Node

Some data sources require collection by using some sort of an API. These APIs can include REST, web services, JMS and/or JDBC as the query mechanism. Splunk as well as third-party developers provide a wide variety of applications that allow these API interactions to occur. Most commonly, these applications are implemented using the Splunk Modular Input framework, which requires a full Splunk enterprise software install to properly function. The best practice to realize this use case is to deploy one or more servers to work as a heavy forwarder configured to work as a Data Collection Node (DCN).

Data Collection Node Topology



(HEC) HTTP Event Collector

The HEC provides a listener service that accepts HTTP/S connections on the server side, and an API on the client side, allowing applications to post log data payloads directly to either the indexing tier or a dedicated HEC receiver tier consisting of one or more heavy forwarders. HEC provides two endpoints that support data to be sent either in raw format or in JSON format. Utilizing JSON can allow for additional metadata to be included in the event payload that may facilitate greater flexibility when searching the data later.

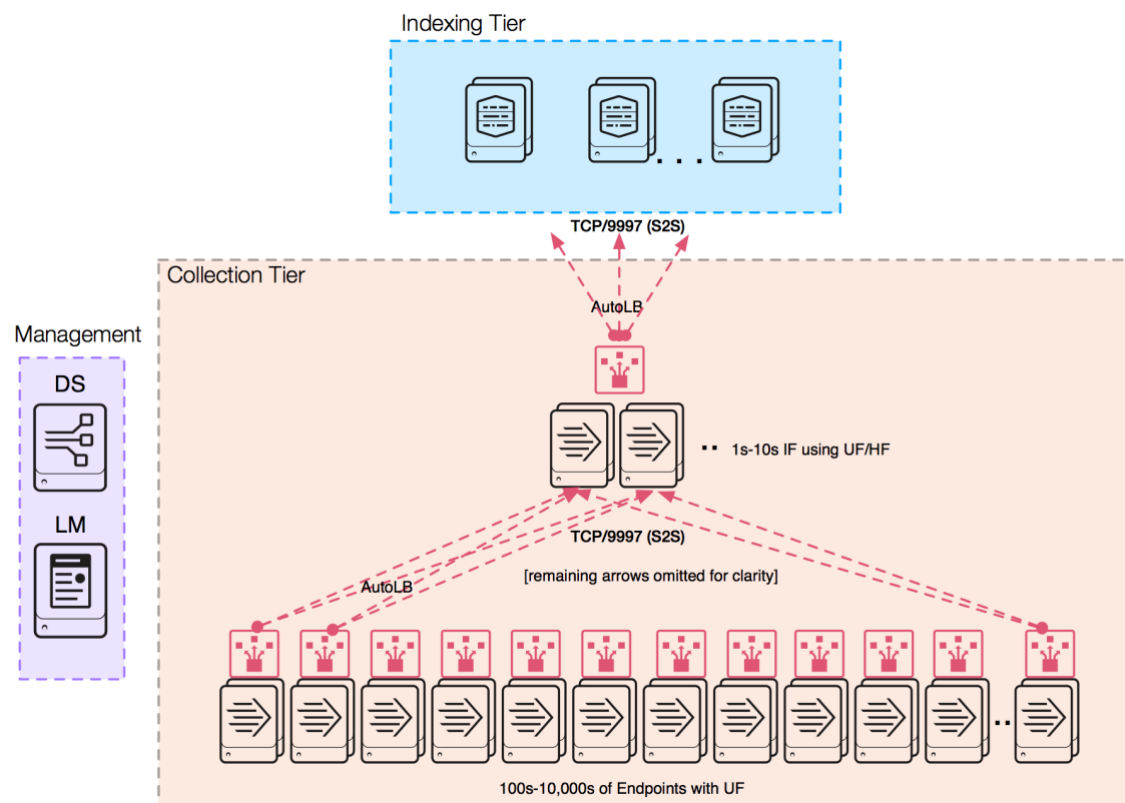
Note: This HEC deployment architecture is providing the transport for some of the other data collection components discussed later, specifically Syslog and metrics data collection.

(IF) Intermediary Forwarding Tier

In some situations, intermediary forwarders are needed for data forwarding. Intermediary forwarders receive log streams from endpoints and forward it on to an indexer tier. Intermediary forwarders introduce architectural challenges that require careful design in order to avoid negative impacts to the overall Splunk environment. Most prominently, intermediary forwarders concentrate connections from 100s to 10,000s of endpoint forwarders and forward to indexers using a far smaller number of connections. This can materially affect the data distribution across the indexing tier, because only a subset of indexers are receiving traffic at any given point in time. However, these negative side effects can be mitigated by proper sizing and configuration.

The following diagram illustrates this challenge well:

Intermediary Forwarding Topology



In a scenario with a single intermediary forwarder, all endpoints connect to this single forwarder (potentially thousands), and the intermediary forwarder in turn only connects to one indexer at any given time. This is not an optimal scenario because the following consequences are likely to occur:

- A large data stream from many endpoints is funneled through a single pipe that exhausts your system and network resources.
- Limited failover targets for the endpoints in case of IF failure (your outage risk is reverse proportional to the number of IFs).
- Small number of indexers are served at any given point in time. Searches over short time periods will not benefit from parallelization as much as they could otherwise.

Intermediary forwarders also add an additional architecture tier to your deployment which can complicate management and troubleshooting and adds latency to your data ingest path. Try to avoid using intermediary forwarding tiers unless this is the only option to meet your requirements. You may consider using an intermediary tier if you have:

- Sensitive data needs to be obfuscated/removed before sending across the network to indexers. An example is when you must use a public network.
- Strict security policies do not allow for direct connections between endpoints and indexers such as multi-zone networks or cloud-based indexers.
- Bandwidth constraints between endpoints and indexers requiring a significant subset of events to be filtered.
- Event-based routing to dynamic targets is requirements.

Consider sizing and configuration needs for any intermediary forwarding tier to ensure availability of this tier, provide sufficient processing capacity to handle all traffic, and support good event distribution across indexers. The IF tier has the following requirements:

- Sufficient number of data processing pipelines overall.
- Redundant IF infrastructure.
- Properly tuned Splunk load-balancing configuration. For example, `autoLBVolume`, `EVENT_BREAKER`, `EVENT_BREAKER_ENABLE`, possibly `forceTimeBasedAutoLB` as needed.

The general guideline suggests to have twice as many IF processing pipelines as indexers in the indexing tier.

Note: A processing pipeline does not equate to a physical IF server. Provided sufficient system resources. For example, CPU cores, memory and NIC bandwidth, are available, a single IF can be configured with multiple processing pipelines.

If you need an IF tier ([see questionnaire](#)), default to using UF for the tier since they provide higher throughput at a lower resource footprint for both the system and network. Use HF if you the UF capabilities do not meet you requirements.

(SYSLOG) Syslog Data Collection

The syslog protocol delivers a ubiquitous source for log data in the enterprise. Most scalable and reliable data collection tiers contain a syslog ingestion component. There are many ways to get syslog data into Splunk. Consider the following methods:

- **Universal forwarder (UF)/heavy forwarder (HF):** Use a Splunk UF or HF to monitor (ingest) files written out by a syslog server (such as `rsyslog` or `syslog-ng`).
- **Syslog Agent to HEC:** Use a syslog agent that is capable to output to Splunk's HEC. (there are third-party modules for `rsyslog` and `syslog-ng` that can output to HEC). .
- **Direct TCP/UDP Input:** Splunk has the ability to listen on a TCP or UDP port (default port is UDP 514) and ingest sources here (**not** recommended for production use).

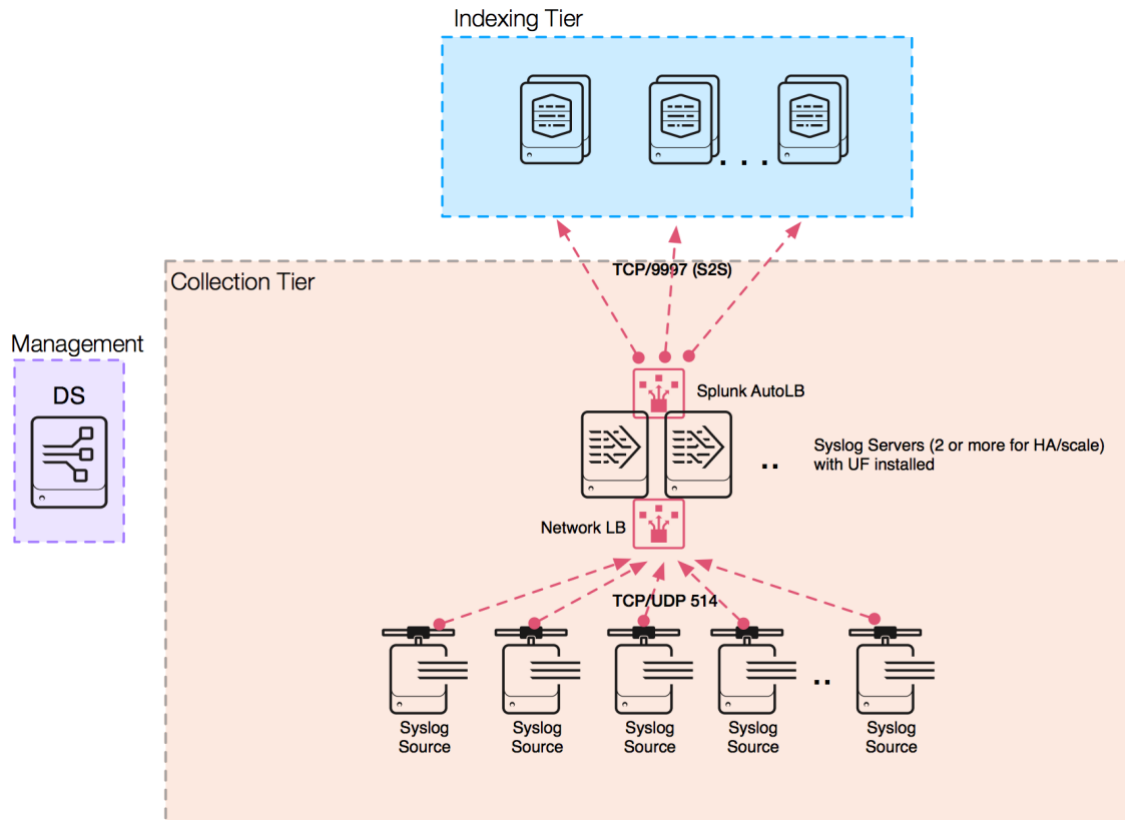
Syslog (File Monitoring in conjunction with a SCD)

Splunk can use monitoring, with `inputs.conf`, on a UF/HF to process and ingest syslog sources that are written to disk on an endpoint by a syslog collection daemon (SCD). Most commonly encountered, both `rsyslog`, `syslog-ng`, and [Fastvue](#) offer commercial and free solutions that are both scalable and simple to integrate and manage in both low volume environments and large scale distributed environments.

To learn more about how to configure monitors, see [Monitor files and directories](#) in *Getting Data In*.

This architecture supports proper data onboarding in the same way a universal forwarder does on any other endpoint. You can configure the SCD to identify multiple different log types and write out log events in appropriate files and directories where a Splunk forwarder can pick them up. This also adds a level of persistence to the syslog log stream by writing events to disk, which can limit exposure to data loss for messages sent using the unreliable UDP as transport.

Syslog Data Collection Topology using UF



The diagram shows syslog sources sending data using TCP or UDP on port 514 to a load-balanced pool of syslog servers. Multiple servers ensure HA for the collection tier and can prevent data loss during maintenance operations. Each syslog server is configured to apply rules to the syslog stream that result in syslog events being written to dedicated files/directories for each source type (firewall events, OS syslog, network switches, IPS, etc.). The UF that is deployed to each server monitors those files and forwards the data to the indexing tier for processing into the appropriate index. Splunk AutoLB is used to distribute the data evenly across the available indexers.

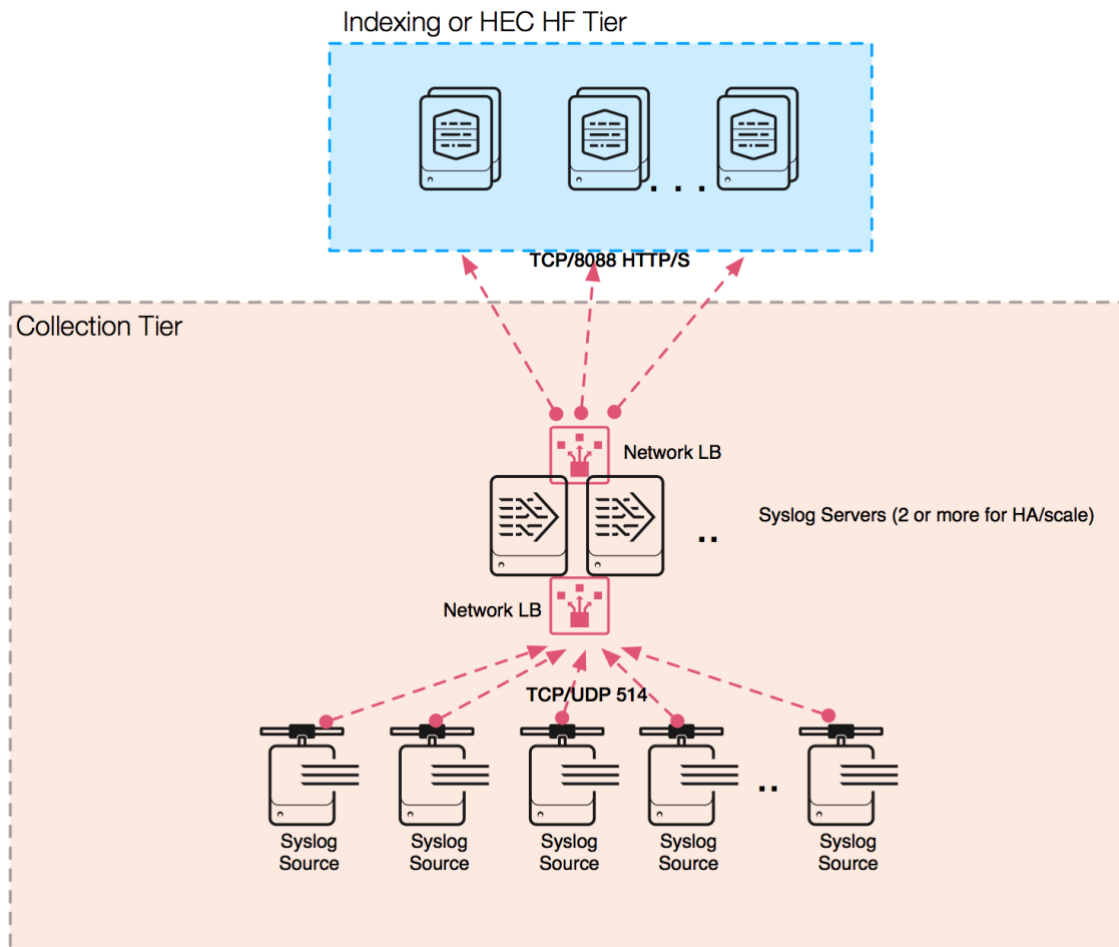
The deployment server shown in the management tier can be used to centrally manage the UF configuration

Syslog Agent to HEC

With the increased adoption of the HEC, there are increasing numbers of deployments that are utilizing their deployment of the HEC to ingest syslog. To learn more, see the Splunk Blogs post [Syslog-ng and HEC: Scalable Aggregated Data Collection in Splunk](#).

The diagram below shows syslog sources sending data on port 514 using a network load balancer to a syslog server farm. Appropriate syslog policies with a custom syslog destination, a python script utilizing the HEC API, get applied and the events are sent to a HEC listener, also with a network traffic load balancer for indexing:

Syslog Data Collection Topology using HEC



A benefit of this topology is it eliminates the need to deploy and configure UF/HF. The HTTP load balancer serves the HEC listeners on the indexers (or a dedicated HEC listener tier) to ensure the data being spread across the HEC endpoints evenly. Configure this load balancer with the "Least Connections" policy.

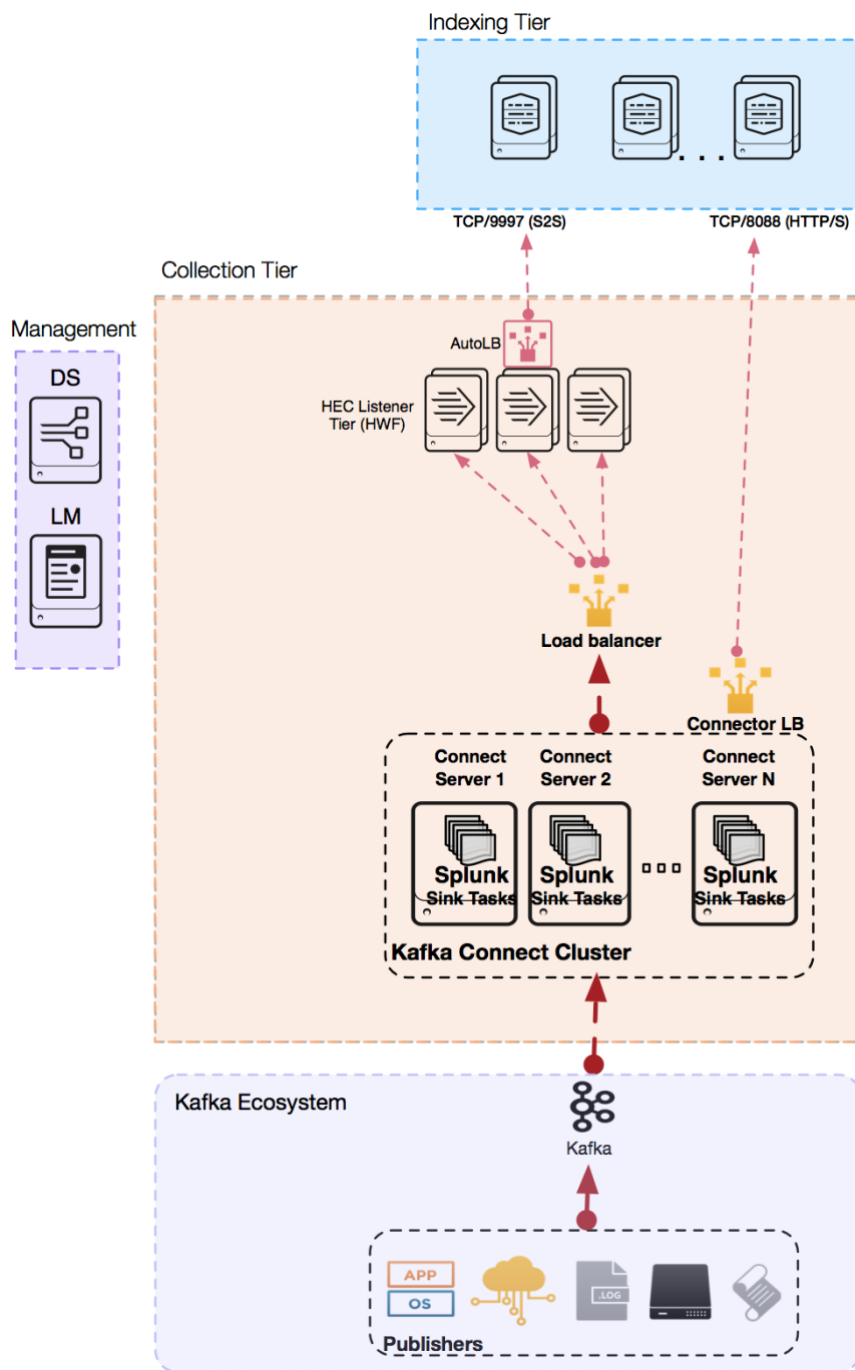
Splunk UDP Input

Splunk can utilize a direct UDP input on a UF or HF to receive data from syslog. To learn about configuring TCP and UDP ports, see [Get data from TCP and UDP ports](#) in *Getting Data In*. The ability to receive events on UDP 514 relies on the ability of the UF/HF to run as root. Additionally, the agent must be available 100% of the time to avoid the possibility of data loss. Forwarders may be restarted frequently to apply configuration changes, which pretty much guarantees data loss. For these reasons, **this is not considered a best practice for a production deployment.**

(KAFKA) Consuming Log Data from Kafka Topics

Splunk provides a supported sink connector for consuming data from Kafka topics called "Splunk Connect for Kafka". See [Apache Kafka Connect](#) in the Splunk Connect for Kafka Manual for detailed product documentation. The Splunk Connect for Kafka package is installed into a properly sized Kafka Connect cluster (outside of Splunk), where it can subscribe to topics as configured and send consumed events using the HEC to be indexed:

Data Collection Topology using Kafka and HEC

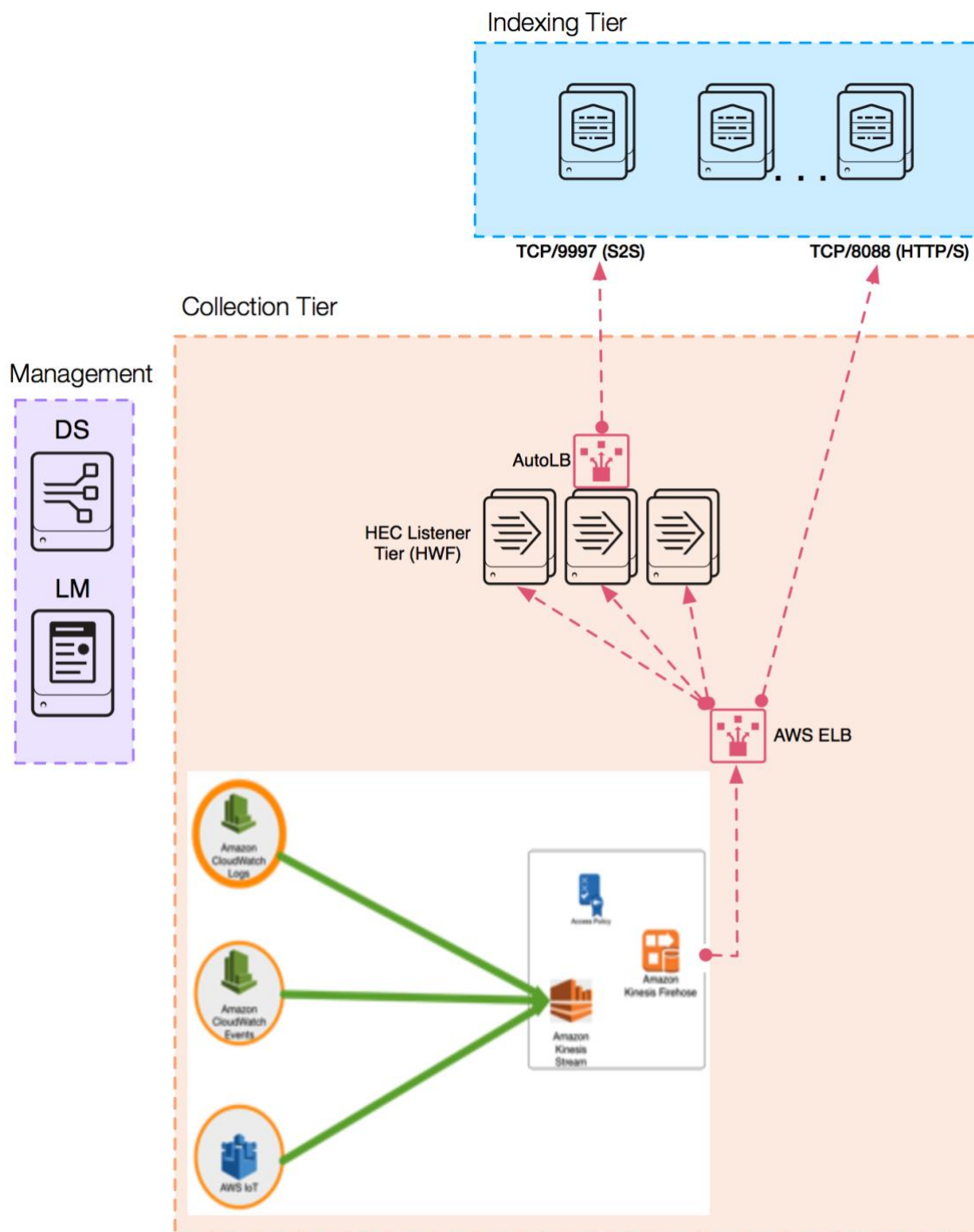


The diagram shows Kafka Publishers sending messages to the Kafka bus. The tasks hosted in the Kafka Connect cluster consume those messages via the Splunk Connect for Kafka and send the data to the HEC listening service using a network load balancer. Again, the HEC listening service can be either hosted directly on the indexers, or on a dedicated HEC listener tier. Please refer to the HEC section for details. Management tier components are only required if a dedicated HF tier is deployed to host HEC listeners.

(KINESIS) Consuming Log Data from Amazon Kinesis Firehose

Splunk and Amazon have implemented an integration between Kinesis and the Splunk HEC that enables you to stream data from AWS directly to a HEC endpoint, configurable via your AWS console. This is complemented by the [Splunk Add-On for Kinesis Firehose](#) which provides CIM-compliant knowledge for various data sources originating in AWS.

Data Collection Topology using Amazon Kinesis



The diagram shows AWS log sources being sent using a Kinesis stream to the Firehose, which — with proper configuration — will send the data to the HEC listening service via an AWS ELB. Again, the HEC listening service can be either hosted directly on the indexers, or on a dedicated HEC listener tier. Please refer to the HEC section for details.

Management tier components shown are only required if a dedicated HF tier is deployed to host HEC listeners.

(METRICS) Metrics Collection

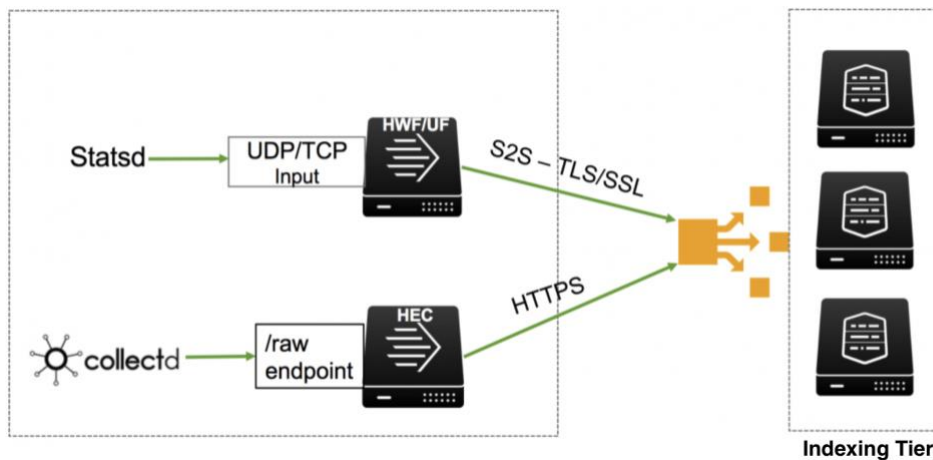
Splunk has the capability to receive and collect system and application performance data, or Metrics data, from a variety of 3rd party software. Metrics in the Splunk platform use a custom index type that is optimized for metric storage and retrieval.

There are different ways to consume metrics data and the collection method is based upon the technology used. The most common form of metrics collections comes in the form of a software daemon, such as **collectd**, **statsd** or using a customized metrics data file and valid configuration for the data source.

There are primarily two methods for getting metrics into Splunk when using agents such **statsd** and **collectd**. Either using a **Direct TCP/UDP** input or via the **HEC**.

Using **HEC** is considered a best practice due to the resiliency and scalability of the **HEC** endpoint, and the ability to horizontally scale the collection tier easily.

Metrics Data Collection Topology



Statsd currently supports **UDP** and **TCP** transport, which you can use as a direct input on a Splunk Forwarder, or Indexer. However, it is not a best practice to send TCP/UDP traffic directly to forwarders in production as the architecture is not resilient and prone to event loss (see Syslog collection) caused by required Splunk forwarder restarts.

(HA) High-Availability Considerations for Forwarding Tier components

There is a common concept of high availability (HA) in the digital world. However, depending upon the organization, the meaning can vary and be more in line with disaster recovery (DR) as opposed to high availability. These two concepts, while similar, do have different meanings. HA is a characteristic of a system, which aims to ensure an agreed level of operational performance, usually uptime, for a higher than normal period. DR involves a set of policies, tools, and procedures to enable the recovery or continuation of vital technology infrastructure and systems following a disaster.

The following outlines various forms of HA at the intermediate/aggregation tier:

Intermediate Tier

- For customers with intermediate/aggregate tier deployments, HA of forwarders is mission critical. At the application layer, Splunk currently does not have a native support method for HA. There are other strategies for providing HA at the operating system level that are not native to Splunk. Common solutions include VMWare VMotion, AWS Autoscaling Groups, and Linux Clustering. Consult with your Splunk Architect to discuss other design options available to you.
- For environments with an HA requirements for a dedicated HEC tier, it's a best practice to use a network traffic load balancer (NTLB), such as NGINX, in front of multiple Splunk heavy forwarders. This provides the advantage of maximum throughput, scale and availability. You have a dedicated pool of HTTP event collector instances whose only job is to receive and forward data. You can add more HEC instances without necessarily having to add more indexers. If your indexers becomes a bottleneck, add additional indexers.
- For environments with an HA requirement for syslog collection, it's a best practice to use multiple Syslog servers served by a cluster (virtual) IP address hosted by a load balancing solution, like HAProxy or F5 to provide maximum throughput, scale and availability. You have a dedicated pool of Splunk instances whose only job is to receive and forward data. You can add more instances without necessarily having to add more indexers. If your indexers become a bottleneck, add additional indexers.

Forwarding Tier

- At the forwarding (endpoint) tier, HA for the agent itself is dependent upon the underlying OS. At the very minimum, you should ensure that any services that implement forwarding functionality are restarted automatically when the host OS restarts. Outside of that, best practices for the forwarders would involve the configuration and proper use of AutoLB from the forwarders to multiple indexers. This also involves use of indexer acknowledgement in order to guarantee data arrives to the indexing tier.

Step 3: Apply Design Principles and Best Practices

Below you will find design principles and best practices separated by deployment tier.

Deployment Tiers

SVA design principles cover all of the following deployment tiers:

Tier	Definition
Search	<ul style="list-style-type: none"> Search heads
Indexing	<ul style="list-style-type: none"> Indexers
Collection	<ul style="list-style-type: none"> Forwarders Modular Inputs Network HEC (HTTP Event Collector) etc
Management / Utility	<ul style="list-style-type: none"> CM DS LM DMS SHC-D

Aligning Your Topology with Best Practices

You will need to keep your requirements and topology in mind in order to select the appropriate design principles and best practices for your deployment. Therefore, you should consider best practices only after you have completed Steps 1 and 2 of the Splunk Validated Architectures selection process above.

Best Practices: Tier-Specific Recommendations

Below you will find design principles and best practices recommendation for each deployment tier. Each design principle reinforces one or more of the SVA pillars: Availability, Performance, Scalability, Security, and Manageability.













Search Tier Recommendations

DESIGN PRINCIPLES / BEST PRACTICES (Your requirements will determine which practices apply to you)		SVA PILLARS				
		AVAILABILITY	PERFORMANCE	SCALABILITY	SECURITY	MANAGEABILITY
1	Keep search tier close (in network terms) to the indexing tier <i>Any network delays between search and indexing tier will have direct impact on search performance</i>		✓			
2	Avoid using multiple independent search heads <i>Independent search heads do not allow sharing of Splunk artifacts created by users. They also do not scale well with respect to resource utilization across the search tier. Unless there is a specific need to have isolated search head environments, there is a better option to scale.</i>	✓		✓	✓	✓
3	Exploit Search Head Clustering when scaling the search tier <i>A search head cluster replicates user artifacts across the cluster and allows intelligent search workload scheduling across all members of the cluster. It also provides a high availability solution.</i>	✓		✓		
4	Forward all search heads' internal logs to indexing tier <i>All indexed data should be stored on the indexing tier only. This removes the need</i>		✓			✓

	to provide high-performing storage on the search head tier and simplifies management. Note: This also applies to any other Splunk roles.					
5	Consider using LDAP auth whenever possible <i>Centrally managing user identities for authentication purposes is a general enterprise best practice, simplifies management of your Splunk deployment and increases security.</i>				✓	✓
6	Ensure enough cores to cover concurrent search needs <i>Every search requires a CPU core to execute. If no cores are available to run a search, the search will be queued, resulting in search delays for the user. Note: Applicable to Indexing Tier as well.</i>	✓	✓	✓		
7	Utilize scheduled search time windows as possible / smooth scheduled search load <i>Often, scheduled searches run at specific points in time (on the hour, 5/15/30 minute after the hour, at midnight). Providing a time window that your search can run in helps avoiding search concurrency hotspots.</i>		✓	✓		
9	Limit the number of distinct search head clusters so as not to overwhelm indexing tier <i>Search workload can only be governed automatically within a SH environment. Independent SHCs have the potential to create more concurrent search workload than the indexer (search peer) tier can handle. The same is true for carefully planning the number of standalone search heads.</i>	✓		✓		
10	When building Search Head Clusters, use an odd number of nodes (3,5,7,etc.) <i>SHC captain election is performed using a majority-based protocol. An odd number of nodes ensures that a SHC can never be split into even numbers of nodes during network failures.</i>	✓				✓

Indexing Tier Recommendations

DESIGN PRINCIPLES / BEST PRACTICES (Your requirements will determine which practices apply to you)		PILLARS				
		AVAILABILITY	PERFORMANCE	SCALABILITY	SECURITY	MANAGEABILITY
1	<p>Enable parallel pipelines on capable servers to</p> <p><i>Parallelization features enable exploitation of available system resources that would otherwise sit idle. Note that I/O performance must be adequate before enabling ingest parallelization features.</i></p>		✓	✓		
2	<p>Consider using SSDs for HOT/WARM volumes and Summaries</p> <p><i>SSDs have reached economical prices and remove any possible IO limitations that are often the cause for unsatisfactory search performance.</i></p>		✓			
3	<p>Keep indexing tier close (in network terms) to the search tier.</p> <p><i>Lowest possible network latency will have positive effect on user experience when searching.</i></p>		✓			
4	<p>Use index replication when historical data / report HA is needed.</p> <p><i>Index replication ensures multiple copies of every event in the cluster to protect against search peer failure. Adjust the number of copies (replication factor) to match your SLAs.</i></p>	✓				
5	<p>Ensure good data onboarding hygiene (e.g. line breaking, timestamp extraction, TZ, and source, source type, host are properly and explicitly defined for each data source) and establish ongoing monitoring using the Monitoring Console.</p> <p><i>Explicitly configuring data sources vs. relying on Splunk's auto-detection capabilities has been proven to have significant benefit to data ingest capacity and indexing latency, especially in high-volume deployments.</i></p>		✓	✓		✓

6	<p>Consider configuring batch mode search parallelization setting on indexers with excess processing power</p> <p><i>Exploiting search parallelization features can have a significant impact on search performance for certain types of searches and allows you to utilize system resources that may otherwise be unused</i></p>					
7	<p>Monitor for balanced data distribution across indexer nodes (=search peers).</p> <p><i>Even event/data distribution across the search peers is a critical contributing factor for search performance and proper data retention policy enforcement.</i></p>					
8	<p>Disable web UI on indexers in distributed/clustered deployments.</p> <p><i>There is no reasonable need to access the WebUI directly on indexers.</i></p>					
9	<p>Consider Splunk pre-built Technology Add-Ons for well-known data sources</p> <p><i>Rather than building your own configuration to ensure data onboarding hygiene for well understood data sources, Splunk-provided TAs can provide faster time to value and ensure optimal implementation.</i></p>					
10	<p>Monitor critical indexer metrics</p> <p><i>Splunk provides you with a monitoring console that provides key performance metrics on how your indexing tier is performing. This includes CPU and memory utilization, as well as detailed metrics of internal Splunk components (processes, pipelines, queues, search).</i></p>					

Collection Tier Recommendations

DESIGN PRINCIPLES / BEST PRACTICES (Your requirements will determine which practices apply to you)		PILLARS				
		AVAILABILITY	PERFORMANCE	SCALABILITY	SECURITY	MANAGEABILITY
1	Use UF to forward data whenever possible. Use of the Heavy Forwarder should be limited to the use cases that require it. <i>Built-in autoLB, restart capable, centrally configurable, small resource demand</i>		✓			✓
2	Use at least 2x intermediary forwarding pipelines to indexers when funneling many UFs <i>Multiplexing a large number of endpoint forwarders across a small number of intermediary forwarders impacts even event distribution across indexers, which affects search performance. Only deploy intermediary forwarders if absolutely necessary.</i>	✓	✓			
3	Consider securing UF-IDX traffic using SSL				✓	
4	Use native Splunk LB to spray data to indexing tier <i>Network load-balancers are not currently supported <u>between forwarders and indexers</u>.</i>	✓		✓		
5	Use dedicated syslog servers for syslog collection <i>Syslog servers can persist TCP/UDP traffic to disk based on source and enable proper sourcetype configuration for processing with a universal forwarder. Required forwarder restarts will not cause data loss.</i>	✓				✓
6	Use HEC for agent-less collection (instead of native TCP/UDP) <i>The HTTP Event Collector (HEC) is a listening service that allows events to be posted via the HTTP[S] protocol. It can be enabled directly on indexers, or configured on a heavy forwarder tier; both served by a load balancer.</i>	✓				✓

Management / Utility Tier Recommendations

DESIGN PRINCIPLES / BEST PRACTICES (Your requirements will determine which practices apply to you)		PILLARS				
		AVAILABILITY	PERFORMANCE	SCALABILITY	SECURITY	MANAGEABILITY
1	Consider consolidating LM, CM, SHC-D and MC on a single instance for small environments <i>These server roles have very little resource demands and are good candidates for colocation. In larger indexer clusters, the CM may require a dedicated server to efficiently manage the cluster.</i>					✓
2	Consider a separate instance for DS for medium to large deployments <i>Once a significant number of forwarders are managed via the Deployment Server, the resource needs will increase to where a dedicated server is required to maintain the service.</i>					✓
3	Consider multiple DSs behind LB for super large deployments <i>Note: This may require help from Splunk professional services to be setup and configured properly</i>	✓		✓		✓
4	Determine whether DS phoneHomeIntervalInSecs can be backed off the 60 second default <i>A longer phone home interval will have positive effect on DS scalability</i>			✓		
5	Use dedicated/secured DS to avoid client exploitation via app deployment <i>Anyone with access to the Deployment Server can modify Splunk configuration managed by that DS, including potentially deploying malicious application to forwarder endpoints. Securing this role appropriately is prudent.</i>				✓	
6	Use the Monitoring Console (MC) to monitor the health of your deployment and alert on health issues. <i>The monitoring console provides a pre-built, splunk-specific set of monitoring solutions and contains extensible platform alerts that can notify you about degrading health of your environment.</i>	✓	✓			✓

Summary & Next Steps

This whitepaper has provided a general introduction to Splunk Validated Architectures. A Validated Architecture ensures that your organization's requirements are being met in the most cost-effective, manageable, and scalable way possible. SVAs offer best practices and design principles built upon the following foundational pillars:

- Availability
- Performance
- Scalability
- Security
- Manageability

This whitepaper has also covered the 3-step Splunk Validated Architectures selection process: 1) Definition of requirements, 2) Choosing a topology, and 3) Applying design principles and best practices. Now that you are familiar with the multiple benefits of Splunk Validated Architectures, we hope you are ready to move forward with the process of choosing a suitable deployment topology for your organization.

Next Steps

So, what comes after choosing a Validated Architecture? The next steps on your journey to a working environment include:

Customizations

- Consider any necessary customizations your chosen topology may need to meet specific requirements.

Deployment Model

- Decide on deployment model (bare metal, virtual, cloud).

System

- Select your technology (servers, storage, operating systems) according to Splunk system requirements.

Sizing

- Gather all the relevant data you will need to size your deployment (data ingest, expected search volume, data retention needs, replication, etc.) [Splunk Storage Sizing](https://splunk-sizing.appspot.com/) (<https://splunk-sizing.appspot.com/>) is one of the available tools.

Staffing

- Evaluate your staffing needs to implement and manage your deployment. This is an essential part of building out a Splunk Center of Excellence.

We are here to assist you throughout the Validated Architectures process and with next steps. Please feel free to engage your Splunk Account Team with any questions you might have. Your Account Team will have access to the full suite of technical and architecture resources within Splunk and will be happy to provide you with further information.

Happy Splunking!

Appendix



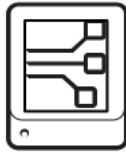
This section contains additional reference information used in the SVAs.

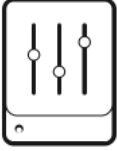





Appendix "A": SVA Pillars Explained

Pillar	Description	Primary Goals / Design Principles
Availability	The ability to be continuously operational and able to recover from planned and unplanned outages or disruptions.	<ol style="list-style-type: none"> 1. Eliminate single points of failure / Add redundancy 2. Detect planned and unplanned failures/outages 3. Tolerate planned/unplanned outages, ideally automatically 4. Plan for rolling upgrades
Performance	The ability to effectively use available resources to maintain optimal level of service under varying usage patterns.	<ol style="list-style-type: none"> 1. Add hardware to improve performance; compute, storage, memory. 2. Eliminate bottlenecks 'from the bottom up' 3. Exploit all means of concurrent processing 4. Exploit locality (i.e. minimize distribution of components) 5. Optimize for the common case (80/20 rule) 6. Avoid unnecessary generality 7. Time shift computation (pre-compute, lazily compute, share/batch compute) 8. Trade certainty and accuracy for time (randomization, sampling)
Scalability	The ability to ensure that the system is designed to scale on all tiers and handle increased workloads effectively.	<ol style="list-style-type: none"> 1. Scale vertically and horizontally 2. Separate functional components that need to be scaled individually 3. Minimize dependencies between components 4. Design for known future growth as early as possible 5. Introduce hierarchy in the overall system design
Security	The ability to ensure that the system is designed to protect data as well as configurations/assets while continuing to deliver value.	<ol style="list-style-type: none"> 1. Design for a secure system from the start 2. Employ state-of-the art protocols for all communications

Pillar	Description	Primary Goals / Design Principles
		<ol style="list-style-type: none"> 3. Allow for broad-level and granular access to event data 4. Employ centralized authentication 5. Implement auditing procedures 6. Reduce attack or malicious use surface area
Manageability	The ability to ensure the system is designed to be centrally operable and manageable across all tiers.	<ol style="list-style-type: none"> 1. Provide a centralized management function 2. Manage configuration object lifecycle (source control) 3. Measure and monitor/profile application (Splunk) usage 4. Measure and monitor system health

Appendix "B": Topology Components

Tier	Component	Icon	Description	Notes
Management	Deployment Server (DS)		The deployment server manages configuration of forwarder configuration.	Should be deployed on a dedicated instance. It can be virtualized for easy failure recovery.
	License Master (LM)		The license master is required by other Splunk components to enable licensed features and track daily data ingest volume.	The license master role has minimal capacity and availability requirements and can be colocated with other management functions. It can be virtualized for easy failure recovery.
	Monitoring Console (MC)		The monitoring console provides dashboards for usage and health monitoring of your environment. It also contains a number of pre-packaged platform alerts that can be customized to provide notifications for operational issues.	In clustered environments, the MC can be colocated with the Master Node, in addition to the License Master and Deployment server function in non-clustered deployments. It can be virtualized for easy failure recovery.

Tier	Component	Icon	Description	Notes
	Cluster Master (CM)		The cluster master is the required coordinator for all activity in a clustered deployment.	In clusters with a large number of index buckets (high data volume/retention), the cluster master will likely require a dedicated server to run on. It can be virtualized for easy failure recovery.
	Search Head Cluster Deployer (SHC-D)		The search head cluster deployer is needed to bootstrap a SHC and manage Splunk configuration deployed to the cluster.	The SHC-D is not a runtime component and has minimal system requirements. It can be colocated with other management roles. Note: Each SHC requires it's own SHC-deployer function. It can be virtualized for easy failure recovery.
Search	Search Head (SH)		The search head provides the UI for Splunk users and coordinates scheduled search activity.	Search heads are dedicated Splunk instances in distributed deployments. Search heads can be virtualized for easy failure recovery, provided they are deployed with appropriate CPU and memory resources.
	Search Head Cluster (SHC)		A search head cluster is a pool of at least three clustered Search Heads. It provides horizontal scalability for the search head tier and transparent user failover in case of outages.	Search head clusters require dedicated servers of ideally identical system specifications. Search head cluster members can be virtualized for easy failure recovery, provided they are deployed with appropriate CPU and memory resources.
Indexing	Indexer		Indexers are the heart and soul of Splunk. They process and index incoming data and also serve as search peers to fulfill search requests initiated on the search tier.	Indexers must always be on dedicated servers in distributed or clustered deployments. In a single-server deployment, the indexer will also provide the search UI and license master functions. Indexers perform best on bare metal servers or in dedicated, high-performance virtual machines, if adequate resources can be guaranteed.
Data Collection	Forwarders and other data collection components		General icon for any component involved in data collection.	This includes universal and heavy forwarders, network data inputs and other forms of data collection (HEC, Kafka, etc.)