Splunk® Enterprise
Capacity Planning Manual 9.0.0

Generated: 7/01/2022 10:20 pm

Table of Contents

Introduction

Introduction to capacity planning for Splunk Enterprise

You can expand Splunk Enterprise to meet almost any capacity requirement. To take advantage of this scaling capability requires planning. This manual discusses high-level hardware guidance for Splunk Enterprise deployments and describes how Splunk Enterprise uses hardware resources in different situations.

New for Splunk Enterprise version 6.2 and later, this manual supersedes guidance about capacity planning in the *Installation* and *Distributed Deployment* manuals. It provides information about reference hardware and a performance checklist to determine when and how you should scale your deployment based on your needs.

Before you decide on hardware for Splunk Enterprise, see the following information:

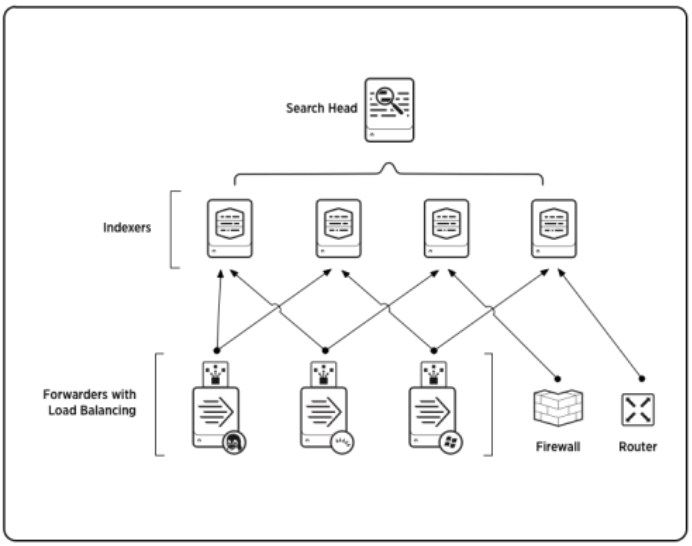
- Review Components of a Splunk Enterprise Deployment in this manual for a description of the elements in a Splunk Enterprise installation.
- Read about the dimensions of a Splunk Enterprise deployment, how those dimensions impact performance, and how to maximize performance.
- Learn about the basic building block of a Splunk Enterprise deployment in Reference hardware in this manual.

Hardware capacity planning

Components of a Splunk Enterprise deployment

The simplest deployment is the one you get by default when you first install Splunk Enterprise on a machine: a standalone instance that handles both indexing and searching. You log into Splunk Web or the CLI on the instance and configure data inputs to collect machine data. You then use the same instance to search, monitor, alert, and report on the incoming data.

You can also deploy specialized instances of Splunk Enterprise on multiple machines to address your load and availability requirements. These specialized instances are called "components". This section introduces the types of components. See the *Distributed Deployment* manual, particularly the topic, Scale your deployment with Splunk Enterprise components.



Indexer

Splunk **indexers** provide data processing and storage for local and remote data and host the primary Splunk data store. See How indexing works in the *Managing Indexers and Clusters* manual for more information.

Search head

A **search head** is a Splunk Enterprise instance that distributes searches to indexers (referred to as "search peers" in this context). Search heads can be either dedicated or not, depending on whether they also perform indexing. Dedicated search heads don't have any indexes of their own, other than the usual internal indexes. Instead, they consolidate and display results that originate from remote search peers.

To configure a search head to search across a pool of indexers, see What is distributed search in the *Distributed Search Manual*

Forwarder

Forwarders are Splunk instances that forward data to remote indexers for data processing and storage. In most cases, they do not index data themselves. See the About forwarding and receiving topic in the *Forwarding Data* manual.

Deployment server

A Splunk Enterprise instance can also serve as a **deployment server**. The deployment server is a tool for distributing configurations, apps, and content updates to groups of Splunk Enterprise instances. You can use it to distribute updates to most types of Splunk components: forwarders, non-clustered indexers, and non-clustered search heads. See About deployment server and forwarder management in the *Updating Splunk Enterprise Instances* manual.

Functions at a glance

Functions	Indexer	Search head	Forwarder	Deployment server
Indexing	x			
Web		x		
Direct search		x		
Forward to indexer			x	
Deploy configurations	x		x	x

Index replication and indexer clusters

An **indexer cluster** is a group of indexers configured to replicate each others' data, so that the system keeps multiple copies of all data. This process is known as **index replication**. By maintaining multiple, identical copies of data, indexer clusters prevent data loss while promoting data availability for searching.

Splunk Enterprise clusters feature automatic failover from one indexer to the next. This means that, if one or more indexers fail, incoming data continues to get indexed and indexed data continues to be searchable.

In addition to enhancing data availability, clusters have other features that you should consider when you are scaling a deployment, for example, a capability to coordinate configuration updates easily across all indexers in the cluster. Clusters also include a built-in distributed search capability. See About clusters and index replication in the *Managing Indexers and Clusters of Indexers* manual.

Dimensions of a Splunk Enterprise deployment

A Splunk Enterprise deployment has many dimensions. These scenarios determine whether a single reference machine can handle indexing and search load.

In some cases, a single reference machine can collect, store, and search data efficiently. In other cases, consider adding machines to your Splunk Enterprise deployment to increase performance. Below is a list of items that can have a significant impact on Splunk Enterprise performance.

- **Amount of incoming data.** The more data you send to Splunk Enterprise, the more time it needs to process the data into events that you can search, report, and generate alerts on.

- **Amount of indexed data.** As the amount of data stored in a Splunk Enterprise index increases, so does the I/O bandwidth needed to store data and provide results for searches.

- **Number of concurrent users.** If more than one person at a time uses an instance of Splunk Enterprise, that instance requires more resources for those users to perform searches and create reports and dashboards.

- **Number of saved searches.** If you plan to invoke a lot of saved searches, Splunk Enterprise needs capacity to perform those searches promptly and efficiently. A higher search count over a given period of time requires more resources.

- **Types of search you use.** Almost as important as the number of saved searches is the types of search that you run against a Splunk Enterprise instance. There are several types of search, each of which affects how the indexer responds to search requests.

- **Whether or not you run Splunk apps.** Splunk **apps** and solutions can have unique performance, deployment, and configuration considerations. If you plan to run apps, consider the resource requirements of the apps the you are using. See the documentation for the app for more information.

How do these dimensions impact overall performance?

While these factors have an impact on the basic sizing requirements of your Splunk Enterprise deployment, addressing each of them individually does not guarantee peak performance gain for the deployment. You must discover through trial how these factors correlate with one another in your specific application.

For example, if your Splunk Enterprise deployment calls for a low amount of indexing but has a high number of concurrent users, it has significantly different resource needs than a setup with a low number of concurrent users and a high amount of daily indexing volume. Additionally, as both user count and amount of indexed data rise, you must distribute the environment across multiple servers to maintain a similar performance level. Search types complicate matters, because some searches strain available CPU resources, while others depend on the speed of the disk subsystem.

When should I scale my Splunk Enterprise deployment?

You must understand how the deployment dimensions described in this topic apply to your specific use case. Answer the following questions, and then refer to the performance checklist in this manual to determine when you should add more hardware resources:

- How much data do you expect to index daily?
- How much data do you need to retain and for how long?
- How many users do you expect to search through the data at any one time?
- Do you plan to use certain specific searches more than once?
- Do you want or need to use a Splunk app to present or manipulate your data?

The key to a well-performing installation is to develop a plan early in the deployment cycle to account for both your initial outlay of hardware resources and the addition of resources when the deployment scales up.

How incoming data affects Splunk Enterprise performance

A reference Splunk Enterprise indexer can index a significant amount of data in a short period of time: over 20 megabytes of data per second or over 1.7 terabytes per day. This level of indexing occurs if the server is doing nothing else but consuming data.

Because Splunk Enterprise instances do more than index, consider this figure the maximum throughput for an indexer. Performance changes depending on the size and amount of incoming data. Larger events slow down indexing performance. As events increase in size, the indexer uses more system memory to process and index them.

If you need more indexing capacity than a single indexer can provide, add indexers into the deployment to account for the increased demand.

See the topics in this chapter to learn how other factors impact this performance figure.

How indexed data affects Splunk Enterprise performance

After Splunk Enterprise consumes data and places it into indexes, those indexes grow and take up disk space. As the indexes grow and available disk space decreases, Splunk Enterprise takes more time to index incoming data because the indexer's disk subsystem takes more time to find space to store the data.

This growth has an impact on search, as well. On a single indexer, disk throughput splits between indexing, which is ongoing, and search requests, which are interrupts based on requests scheduled by users. As indexes grow, search slows down because the disk subsystem needs to account for search requests, and it also needs to handle increasingly longer requests to store incoming data. Depending on the type of search, those kinds of requests can be I/O-intensive.

How concurrent users affect Splunk Enterprise performance

A reference indexer needs to dedicate one of its available CPU cores for every search that a user invokes for as long as the search is active. If multiple users are logged in and running searches, the number of available CPU cores can be exhausted quickly.

These figures assume that CPUs are idle when they receive a login or search request. This does not account for other system requests or CPU cores used by Splunk Enterprise to index data. If they are processing any other system requests, then the load splits across other available CPUs.

As CPU cores are used up, all activities on an indexer slow down as the computer splits processing time between indexing, search, and handling on-line users. Only additional indexers can increase capacity for all three functions of Splunk Enterprise operation.

How saved searches / reports affect Splunk Enterprise performance

On a reference indexer, a **saved search** or report consumes about 1 CPU core and a specified amount of memory while it executes. It behaves like an ad-hoc search. A saved search also increases the amount of disk I/O temporarily as the disk subsystem looks through the **indexes** to fetch data.

Each additional saved search that executes at the same time consumes an additional CPU core. This consumption is separate from CPU usage from the operating system and Splunk Enterprise indexing and storage processes.

If more saved searches execute than can be accepted for processing, they wait in a queue until they can be serviced. Splunk Enterprise also warns you when the system reaches the maximum number of queued saved searches. When searches queue up, search results return more slowly.

Adding **search heads** provides additional CPU cores to run more concurrent searches. Adding indexers helps scale with the increased search load and concurrency that comes from adding search heads. Adding RAM to your existing machines helps with concurrent searches but does not give you additional search capacity.

How search types affect Splunk Enterprise performance

You can invoke four types of searches against data stored in a Splunk Enterprise index. Each search type impacts the indexer in a different way.

The following table summarizes the different search types. For dense and sparse searches, Splunk Enterprise measures performance based on number of matching events. With super-sparse and rare searches, performance is measured based on total indexed volume.

Search type	Description	Ref. indexer throughput	Performance impact

Dense	Returns a large percentage (10% or more) of matching results for a given set of data in a given period of time. Dense searches usually tax a server's CPU first, because of the overhead required to decompress the raw data stored in a Splunk Enterprise index. Examples of dense searches include searches that use nothing but a wildcard character, or searching any index.
	
	Examples:
	
	`*`
	
	`index=m â ¦| stats count by fieldA`
	
	`index=a sourcetype=b â ¦| timechart count by myfield`
		Up to 50,000 matching events per second.	CPU-bound

Sparse	Returns a smaller amount of results for a given set of data in a given period of time (anywhere from .01 to 1%) than do dense searches.	Up to 5,000 matching events per second.	CPU-bound
Super-sparse	Returns a small number of results from each index **bucket** that matches the search. A super-sparse search is I/O intensive because the indexer must look through all of the buckets of an index to find the results. If you have a large amount of data stored on your indexer, there are a lot of buckets, and a super-sparse search can take a long time to finish.	Up to 2 seconds per index bucket.	I/O bound
Rare	Similar to a super-sparse search, but receives assistance from **Bloom filters**, which help eliminate index buckets that do not match the search request. Rare searches return results anywhere from 20 to 100 times faster than does a super-sparse search.	From 10 to 50 index buckets per second.	I/O bound

How Splunk apps affect Splunk Enterprise performance

A single Splunk Enterprise indexer can run multiple apps simultaneously. Splunk Enterprise includes several apps which it runs at the same time.

However, the more complex apps offer advanced views that require the use of summarizing and accelerating searches that run in the background. The more background processing an app needs, the more likely you must distribute the processing load across multiple machines.

Many apps require a distributed Splunk Enterprise deployment by design. Whether it is a case of universal forwarders fetching data and sending it to a single central instance, or many indexers and search heads connected together and serving up reports, dashboards, or alerts, Splunk apps often need more than one server to realize both maximum performance and potential in the enterprise.

How Splunk apps affect resource requirements

If you use a Splunk app or solution that gets knowledge by executing a large number of saved searches, then you can overwhelm a single-server Splunk Enterprise instance. Multiple searches quickly exhaust available CPU resources on an indexer. See Accommodate many simultaneous searches in this manual.

When you install an app or solution, read the system requirements outlined in that app or solution's documentation. If the information is not available, contact the authors of the app or solution to get information about what you need to run the app properly.

How Splunk Enterprise calculates disk storage

At a high level, Splunk calculates total disk storage as follows:

```
( Daily average indexing rate ) x ( retention policy ) x 1/2
```

Splunk Enterprise stores raw data at up to approximately half its original size with compression. On a volume that contains 500GB of usable disk space, you can store nearly six months' worth of data at an indexing rate of 5GB/day or ten days' worth at a rate of 100GB/day.

If you need additional storage, you can opt for either more local disks, which is required for frequent searching, or you can use attached or network storage, which is acceptable for occasional searching. Low-latency connections over NFS or SMB/CIFS (Server Message Block/Common Internet File System) are acceptable for searches over long time periods where instant search returns can be compromised to lower cost per GB.

Important: Shares mounted over a Wide Area Network (WAN) connection or on standby storage such as tape are never suitable storage choices for Splunk Enterprise operations.

Estimate your storage requirements

When ingesting data into Splunk Enterprise, the indexing process creates a number of files on disk. The rawdata file contains the source data as events, stored in a compressed form. The index or TSIDX files contain terms from the source data that point back to events in the rawdata file. Typically, the rawdata file is 15% the size of the pre-indexed data, and the TSIDX files are approximately 35% of the size of the pre-indexed data. When you combine the two file sizes, the

rawdata and TSIDX represent approximately 50% of pre-indexed data volume.

The guidance for allocating disk space is to use your estimated license capacity (data volume per day) with a 50% compression estimate. The compression estimates for data sources vary based upon the structure of the data and the fields in the data. Most customers will ingest a variety of data sources and see an equally wide range of compression numbers, but the aggregate compression used to estimate storage is still 50% compression.

For example, to keep 30 days of data in a storage volume at 100GB/day in data ingest, plan to allocate at least (100*30/2) 1.5TB of free space. If you have multiple indexers, you will divide the free space required between all indexers equally. For example, if you have 2 indexers, each indexer needs (100*30/2)/2 750GB of free storage space. The calculation example does not include extra space for OS disk space checks, minimum space thresholds set in other software, or any other considerations outside of Splunk Enterprise.

Planning the index storage

Planning for index storage capacity is based upon the data volume per day, the data retention settings, the number of indexers, and which features of Splunk Enterprise you are using:

- You have the data volume per day estimate used to calculate your license volume.
- You know how long you need to keep your data.
- You have an estimate of how many indexers you need.
- (Optional) You know which data is most valuable to you, and you know how long that data is valuable for.
- (Optional) You know that some data has historical value, but might not need to be searched as often or as quickly.
- (Optional) You have an audit requirement to keep a copy of some data for a period of time, but you plan to restore the data before searching it.
- (Optional) You have verified how well your data compresses. See Use a data sample to calculate compression.
- (Optional) You plan to implement an index cluster. An index cluster requires additional disk space calculations to support data availability. See Storage requirement examples in the *Managing Indexers and Clusters of Indexers* manual.
- (Optional) You plan to implement SmartStore remote storage. See About SmartStore in the *Managing Indexers and Clusters of Indexers* manual.
- (Optional) You plan to implement the Enterprise Security app. See Data model acceleration storage and retention in the Enterprise Security *Installation and Upgrade Manual*.

Splunk Enterprise offers configurable storage tiers that allow you to use different storage technologies to support both fast searching and long-term retention. See How data ages in the *Managing Indexers and Clusters of Indexers* manual.

Use a data sample to calculate compression

Use sample data and your operating system tools to calculate the compression of a data source.

For *nix systems

On *nix systems, follow these steps:

1. Select a data source sample and note its size on disk.
2. Index your data sample using a file monitor or one-shot
3. On the command line, go to `$SPLUNK_HOME/var/lib/splunk/defaultdb/db`.
4. Run `du -ch hot_v*` and look at the last `total` line to see the size of the index.

5. Compare the sample size on disk to the indexed size.

For Windows systems

On Windows systems, follow these steps:

1. Download the du utility from Microsoft TechNet.
2. Extract `du.exe` from the downloaded ZIP file and place it into your `%SYSTEMROOT%` or `%WINDIR%` folder. You can also place `du.exe` anywhere in your `%PATH%`.
3. Select a data source sample and note its size on disk.
4. Index your data sample using a file monitor or one-shot
5. Open a command prompt and go to `%SPLUNK_HOME%\var\lib\splunk\defaultdb\db`.
6. Run `del %TEMP%\du.txt & for /d %i in (hot_v*) do du -q -u %i\rawdata | findstr /b "Size:" >> %TEMP%\du.txt`.
7. Open the `%TEMP%\du.txt file`. You will see `Size: n`, which is the size of each `rawdata` directory found.
8. Add these numbers together to find out how large the compressed persisted raw data is.
9. Run `for /d %i in (hot_v*) do dir /s %i`, the summary of which is the size of the index.
10. Add this number to the total persistent raw data number.

This is the total size of the index and associated data for the sample you have indexed. You can now use this to extrapolate the size requirements of your Splunk Enterprise index and `rawdata` directories over time.

Answers

Have questions? Visit Splunk Answers to see what questions and answers other Splunk users had about data sizing.

Scale your Splunk Enterprise Deployment

Distribute indexing and searching

This topic discusses the reasons to distribute the components of your Splunk platform deployment.

Concepts of distributed indexing and searching

Designing a scalable architecture for the Splunk platform requires knowledge of the Splunk instance roles, and how they were intended to scale.

The two most common roles are the search head and the indexer. They represent the roles that carry the burden of managing user objects, searching, parsing, and data storage.

A **Search head** is responsible for:

- Hosting users.
- Storing user created objects.
- Scheduling searches and alerting.
- Providing visual feedback through dashboards and views.
- Enforcing access controls.

An **Indexer** is responsible for:

- Accepting data streams from forwarders.
- Parsing the data.
- Writing the data into buckets.
- Maintaining the buckets.
- Accepting search requests from the search heads.
- Searching the buckets and streaming results back to the search head.

A search head's tasks are primarily CPU bound. As more users and more apps are added to a search head, the concurrent search load climbs quickly and hits a limit. The limit represents the aggregate search load across all users and apps to a search head's CPU cores.

Adding search heads to the deployment increases the aggregate CPU resources, increasing the aggregate search concurrency and the number of active users and apps supported in the environment.

An indexer's tasks are primarily I/O bound. As more forwarders are added to the network, there are more concurrent data streams to accept, and more data to parse before writing. In addition, the search requests require I/O access and processor time to analyze, collect, and return the requested data. As the data streams increase in volume and the concurrent search requests climb higher, the indexer hits a limit. The limit represents the aggregate search load from all search heads and indexing load from forwarders to an indexer's I/O capacity.

Adding indexers to the deployment increases the total aggregate I/O capacity and storage available to save data, reduces the data volume per indexer load, and reduces the impact of the search load by spreading it across more indexers.

Scaling the Splunk platform

A typical Splunk platform deployment plan is based upon 2 points: indexed data volume per day, and estimated search load. User counts are often used as a proxy for search load. For example, one active user with admin-level search concurrency can sustain the same load on the Splunk platform deployment as several users at a lower role level.

Most Splunk implementations are built around a handful of users and a few apps searching hundreds of gigabytes of data. In that scenario, adding indexers is the preferred method of scaling. The same rule applies when implementing a indexer cluster.

As the search head gains more users, the CPU limitations will become apparent as searches may skip, and users experience slower search result speed. Adding another search head to your distributed deployment does not guarantee improved search performance. As the user count increases, indexers must be added to maintain search performance. For a table with scaling guidelines, see Summary of performance recommendations in this manual.

When planning for a user count at 50 or more, consider implementing a search head cluster to absorb a high level of users while adding redundancy to the search tier.

Scaling performance

As your indexers consume data, they store it in **buckets**, which are the individual elements of an index. As more data comes in, the number of buckets increases. As the number of buckets increases, the indexer must manage the buckets by "rolling" them to make room for new incoming data. This procedure takes up I/O cycles, which reduces the resources available to fetch events for search requests. The impact is noticeable for index buckets that hold smaller amounts of data.

Avoid configuring many indexes comprised of small buckets. For examples utilizing the `maxDataSize` bucket setting, see `indexes.conf.example` in the Splunk Enterprise *Admin* manual.

The number and types of search also impact indexer performance. Most search types leverage an indexer's disk subsystem, but a few will use more CPU. For information on simultaneous searches, see Accommodate concurrent users and searches in this manual.

If the hardware allocated for indexers exceeds the reference machine specifications, consider reviewing and implementing one of the Parallelization settings to improve the performance for specific use cases.

Use the monitoring console to monitor and track resource usage across the Splunk platform environment. For more details, see About the monitoring console in *Monitoring Splunk Enterprise*.

How concurrent users and searches impact performance

The largest performance factor in a Splunk Enterprise deployment are:

- The number of concurrent users.
- The number of concurrent searches.
- The types of searches used.

A user that submits a search request will use one CPU core on each indexer until the search is complete. Any additional searches that the user submits also account for one CPU core. You can adjust the number of global concurrent searches

that a machine can run. See Expected performance and known limitations of real-time searches and reports in the Splunk Enterprise *Search* manual.

The type of search a user invokes also impacts hardware resource usage. See How search types affect Splunk Enterprise performance.

How to maximize search performance

To accommodate the resource overhead of running many concurrent searches, add additional indexers, and maximize the physical memory available to the indexers. The indexers do the bulk of the work in search operations, such as identifying the data requested, reading the data from disk, decompressing it, filtering the data, and streaming the results.

For example, if a search uses 200MB of memory, and there are 48 concurrent search requests, that equates to about 10GB of memory to meet the search load not including the OS requirements. The amount of available memory is an important resource to monitor. While performance on an indexer declines gradually with increased CPU usage from concurrent search jobs, it drops dramatically when all available physical memory is exhausted.

Search performance: A basic scenario

The aggregate run time of all searches increases as the number of available CPU cores on the indexer decreases. For example, on an indexer with no load and 12 available cores, the first searches to arrive for processing can complete within a short period of time. For this scenario, all searches will run to completion within 10 seconds.

12 concurrent searches: One indexer with 12 cores and no data being indexed.

No. of concurrent searches	/ No. of avail. cores	= No. of searches per core	No. of sec. per individual search	= Approx. time (sec.) to complete all searches
12	12	1	10	10

When there are 48 searches to run concurrently, the total time to complete all searches increases significantly.

48 concurrent searches: One indexer with 12 cores and no data being indexed.

No. of concurrent searches	/ No. of avail. cores	= No. of searches per core	No. of sec. per individual search	= Approx. time (sec.) to complete all searches
48	12	4	10	40

Because the indexers do the bulk of the work in search operations, such as identifying the data requested, reading the data from disk, decompressing it, filtering some data, and streaming the results, it is best practice to add indexers to decrease the total amount of time to return all search results.

A deployment with more indexers provides an aggregate increase in cores to improve the search completion time when there are many concurrent searches. When there are more cores available than concurrent searches, the cores can be used by Splunk Enterprise to perform maintenance operations, or can remain idle.

12 concurrent searches: 4 indexers with 12 cores per indexer and no data being indexed.

No. of concurrent searches	/ No. of avail. cores	= No. of searches per core	No. of sec. per individual search	= Approx. time (sec.) to complete all searches
12	48		10	10

No. of concurrent searches	/ No. of avail. cores	= No. of searches per core	No. of sec. per individual search	= Approx. time (sec.) to complete all searches
		1. By default, a search cannot take advantage of multiple cores.		

48 concurrent searches: 4 indexers with 12 cores per indexer and no data being indexed.

No. of concurrent searches	/ No. of avail. cores	= No. of searches per core	No. of sec. per individual search	= Approx. time (sec.) to complete all searches
48	48	1	10	10

Search performance: Indexing data scenario

In an active deployment, the system is not sitting idle while searches arrive. If an indexer ingests 150GB/day of data, then it will use up to 4 of the available cores for indexing processes. With fewer cores available, the time it takes to return all search results increases.

12 concurrent searches: One 12-core indexer, with 8 available cores.

No. of concurrent searches	/ No. of avail. cores	= No. of searches per core	No. of sec. per individual search	= Approx. time (sec.) to complete all searches
12	8	2, as each core will remain in use until the prior search completes.	10	20

48 concurrent searches: One 12-core indexer, with 8 available cores.

No. of concurrent searches	/ No. of avail. cores	= No. of searches per core	No. of sec. per individual search	= Approx. time (sec.) to complete all searches
48	8	6	10	60

48 concurrent searches: 4 12-core indexers with 8 available cores.

No. of concurrent searches	/ No. of avail. cores	= No. of searches per core	No. of sec. per individual search	= Approx. time (sec.) to complete all searches
48	32	2, as each core will remain in use until the prior search completes.	10	20

Having fewer indexers with greater core counts per indexer can decrease the amount of total time for search, but also reduces scaling efficiency by providing less aggregate IOPS for search operations.

Adding indexers reduces the indexing load on any one machine. Additionally, you reduce the search time, lower the impact of high search concurrency, and lower the impact of resource contention for I/O and memory.

Performance Reference

Reference hardware

The reference hardware specification is a baseline for scoping and scaling the Splunk platform for your use. The recommendations are based upon the Splunk Validated Architectures (SVA) white paper on splunk.com.

Reference host specification for single-instance deployments

This represents the minimum basic instance specifications for a production grade Splunk Enterprise deployment. A single-instance represents an S1 architecture in SVA:

- An x86 64-bit chip architecture
- 12 physical CPU cores, or 24 vCPU at 2Ghz or greater speed per core.
- 12GB RAM.
- For storage, review the Indexer recommendation in What storage type should I use for a role?
- A 1Gb Ethernet NIC, optional 2nd NIC for a management network .
- A 64-bit Linux or Windows distribution. See Supported Operating Systems in the *Installation Manual*.

If you are planning a single instance Splunk Enterprise installation and want additional headroom for search concurrency or more Splunk Apps, consider using the indexer mid-range or high-performance specifications described below. Once you've exceeded the ability of a single instance deployment to meet your search and data ingest load, review the distributed deployment models defined in SVA.

Reference host specifications for distributed deployments

Distributed deployments are designed to separate the index and search functionality into dedicated tiers that can be sized and scaled independently without disrupting the other tier. The daily data ingest volume and the concurrent search volume are the two most important factors used when estimating the hardware capabilities and node counts for each tier. The search and indexing roles prioritize different compute resources. The indexing tier uses high-performance storage to store and retrieve data efficiently. The search tier uses CPU cores and RAM to handle ad-hoc and scheduled search workloads.

An increase in search tier capacity corresponds to increased search load on the indexing tier, requiring scaling of the indexer nodes. Scaling either tier can be done vertically by increasing per-instance hardware resources, or horizontally by increasing the total node count. For assistance with sizing a production Splunk Enterprise deployment, contact your Splunk Sales team for guidance with meeting the infrastructure requirements and total cost of ownership.

Search head

A search head uses CPU resources more consistently than an indexer, but does not require the same storage capacity. A search request uses up to 1 CPU core while the search is active. You must account for scheduled searches when you provision a search head in addition to ad-hoc searches that users run. More active users and higher concurrent search loads require additional CPU cores.

Minimum search head specification

- An x86 64-bit chip architecture.
- 16 physical CPU cores, or 32 vCPU at 2Ghz or greater speed per core.
- 12GB RAM.

- For storage, see What storage type should I use for a role?
- A 1Gb Ethernet NIC, optional 2nd NIC for a management network.
- A 64-bit Linux or Windows distribution. See Supported Operating Systems in the *Installation Manual*.

For a review on how searches are prioritized, see the topic Configure the priority of scheduled reports in the *Reporting Manual*. For information on scaling search performance, see How to maximize search performance.

Indexer

When you distribute the indexing process among many indexers, the Splunk platform can scale to consume terabytes of data in a day. Adding indexers distributes the work of search requests and data indexing across all of the indexers. This horizontal scaling of indexers increases performance significantly.

Minimum indexer specification

- An x86 64-bit chip architecture.
- 12 physical CPU cores, or 24 vCPU at 2GHz or greater per core.
- 12GB RAM.
- For storage, see What storage type should I use for a role?
- A 1Gb Ethernet NIC, with optional second NIC for a management network.
- A 64-bit Linux or Windows distribution. See Supported Operating Systems in the *Installation Manual*.

Mid-range indexer specification

This specification adds additional cores and RAM to provide overhead for additional search concurrency in a distributed Splunk Enterprise deployment:

- An x86 64-bit chip architecture.
- 24 physical CPU cores, or 48 vCPU at 2GHz or greater speed per core.
- 64GB RAM.
- For storage, see What storage type should I use for a role?
- A 1Gb Ethernet NIC, with optional second NIC for a management network.
- A 64-bit Linux or Windows distribution. See Supported Operating Systems in the *Installation Manual*.

High-performance indexer specification

This specification adds additional cores, RAM, and storage performance to use for improving indexing throughput and providing overhead for additional search concurrency for use cases where sustained search performance is critical, such as Premium Splunk apps.

- An x86 64-bit chip architecture.
- 48 physical CPU cores, or 96 vCPU at 2GHz or greater speed per core.
- 128GB RAM.
- For storage, see What storage type should I use for a role?
- A 1Gb Ethernet NIC with optional second NIC.
- A 64-bit Linux or Windows distribution. See Supported Operating Systems in the *Installation Manual*.

Recommended hardware for management components

A Splunk Enterprise distributed deployment requires several management components. These components often run on their own instances, and can include:

- Deployment Server
- Heavy Forwarders
- Indexer Cluster Management node
- License Manager
- Monitoring Console
- Search head cluster deployer

When allocating resources for the management components, begin with the reference host specification for single-instance deployments noted above, and adjust the resource allocation to accommodate the scale of your deployment. For detailed sizing and resource allocation recommendations, contact your Splunk account team.

For guidance on management components sharing the same instance based on utilization, see Whether to colocate management components in the *Distributed Deployment Manual*.

If you're using heavy forwarders in an intermediate forwarding tier, and have available resources, you can configure multiple pipelines to improve data distribution. See Configure a forwarder to handle multiple pipeline sets in the *Forwarder Manual*.

What storage type should I use for a role?

Insufficient storage I/O is the most commonly encountered limitation in a Splunk software infrastructure. For best results, review the recommended storage types before provisioning your hardware. For guidance on testing your storage system, see How to test my storage system using FIO on Splunk Answers.

Role	Recommended storage type	Notes
Search Head	SSD, HDD	Search heads with a high ad-hoc or scheduled search loads should use SSD. A HDD-based storage system must provide no less than 800 sustained IOPS. A search head requires at least 300GB of dedicated storage space.
Indexer: Hot and warm index storage, data model storage	SSD	The indexer role requires high performance storage for writing and reading (searching) the hot and warm **index buckets**. The storage volume path is the same for hot and warm buckets, and data model acceleration storage by default.
Indexer: SmartStore	NVMe or SSD, and access to a remote object store	SmartStore is a hybrid storage technology that utilizes high performance local storage for both short-term reads and writes, and as a bucket retrieval cache from cloud-hosted storage. For more information on SmartStore, see SmartStore advantages in the *Managing Indexers and Clusters of Indexers* manual.
Indexer: Cold index storage	HDD, SAN, NAS, Network file systems	A cold index bucket is data that has reached a space or time limit, and is rolled from warm. The cold index can have a unique storage volume path. The cold index buckets are often placed on slower, cheaper storage depending upon the search use case. Storage performance affects how quickly search results, reports, and alerts are returned. An unreliable cold storage volume can impact indexing operations.
Indexer: Frozen bucket storage	SAN, NAS, Network file systems, HDD	A frozen index bucket is data that has reached a space or time limit, and is moved from cold to an archival state. Frozen data can have a unique storage volume path. A frozen index bucket is deleted by default. See Archive indexed data in the *Managing Indexers and Clusters of Indexers* manual.

Notes about optimizing Splunk software and storage usage

- The storage volume where Splunk software is installed must provide no less than 800 sustained IOPS.
- Always configure your index storage to use a separate volume from the operating system. The volume used for the operating system or its swap file is not recommended for Splunk Enterprise data storage. For more information on how indexes are stored, including information on database bucket types and how Splunk stores and ages them, see How the indexer stores indexes in the *Managing Indexers and Clusters of Indexers* manual.
- Always monitor storage availability, bandwidth, and capacity for your indexers. The storage volumes or mounts used by the indexes must have some free space at all times. Storage performance decreases as available space decreases. By default, indexing will stop If the volume containing the indexes goes below 5GB of free space.
- Never store the hot and warm buckets of your indexes on network volumes. Network latency will dramatically decrease indexing performance. You can use network shares such as Distributed File System (DFS) volumes or Network File System (NFS) mounts for the cold index buckets. Searches that include data stored on network volumes will be slower.

Ratio of indexers to search heads

The aggregate search and indexing load determines what Splunk instance role (search head or indexer) the infrastructure needs to scale to maintain performance. For a table with scaling guidelines, see Summary of performance recommendations.

Network latency limits for clustered deployments

A Splunk environment with search head or indexer clusters must have fast, low-latency network connectivity between clusters and cluster nodes. This is particularly important in environments that are planning for multi-site clusters.

- For indexer cluster nodes, network latency should not exceed 100 milliseconds. Higher latencies can significantly slow indexing performance and hinder recovery from cluster node failures.
- For search head clusters, latency should not exceed 200 milliseconds. Higher latencies can impact how fast a search head cluster elects a cluster captain.

Impact of network latency on clustered deployment operations.

Network latency	Cluster Index time. 1 TB of data	Cluster node recovery time
< 100 ms	6202 s	143 s
300 ms	6255 s (+ 1%)	1265 s (+ 884%)
600 ms	7531 s (+ 21%)	3048 s (+ 2131%)

Confirm with your network administrator that the networks used to support a clustered Splunk environment meet or surpass the latency guidelines.

Premium Splunk app requirements

Premium Splunk apps can demand greater hardware resources than the reference specifications in this topic provide. Before architecting a deployment for a premium app, review the app documentation for additional scaling and hardware recommendations. The following list shows examples of some premium Splunk apps and their recommended hardware specifications.

- Splunk Enterprise Security
- Splunk IT Service Intelligence
- Splunk App for PCI

Virtualized Infrastructures

Splunk supports use of its software in virtual hosting environments:

- A hypervisor (such as VMware) must be configured to provide reserved resources that meet the hardware specifications above. An indexer in a virtual machine can consume data about 10 to 15 percent more slowly than an indexer hosted on a bare-metal machine. Search performance in a virtual hosting environment is similar to bare-metal machines.
- The storage performance that a virtual infrastructure provides must account for resource contention with any other active virtual hosts that share the same hardware or storage array. It also must provide sufficient IOPS per instance of a Splunk role. For example, a shared storage array providing SSD-level performance for 10 indexers would require 40000 concurrent IOPS (4000 IOPS x 10 indexers) to service the indexers alone, while simultaneously providing additional IOPS to support any other workloads using the same shared storage.

Splunk Cloud Platform

Splunk offers its machine data platform and licensed software as a subscription service called Splunk Cloud Platform. When you subscribe to the service, you purchase a capacity to index, store, and search your machine data. Splunk Cloud Platform abstracts the infrastructure specification from you and delivers high performance on the capacity you have purchased.

To learn more about Splunk Cloud Platform, visit the Splunk Cloud Platform website.

Self-managed Splunk Enterprise in the cloud

Running Splunk Enterprise in the cloud is another alternative to running it on-premises using bare-metal hardware.

If you run Splunk Enterprise on an Cloud-managed infrastructure:

- Cloud vendors assign processor capacity in virtual CPUs (vCPUs). The vCPU is a logical CPU core, and might represent only a small portion of a CPU's full performance. The classification of a vCPU is determined by the cloud vendor.
- Storage options offered by cloud vendors vary dramatically in performance and price. To maintain consistent search and indexing performance, see the storage type recommendations in What storage type should I use for a role?.

Considerations for deploying Splunk software on partner infrastructure

Many hardware vendors and cloud providers have worked to create reference architectures and solution guides that describe how to deploy Splunk Enterprise and other Splunk software on their infrastructure. For your convenience, Splunk maintains a separate page where Splunk Technology Alliance Partners (TAP) may submit reference architectures and solution guides that meet or exceed the specifications of the documented reference hardware standard. See the Splunk Partner Solutions page on the Splunk website.

While Splunk works with TAPs to ensure that their solutions meet the standard, it does not endorse any particular hardware vendor or technology.

Determine when to scale your Splunk Enterprise deployment

Before you consider when and how to scale your environment, estimate how much data you need to index, and how many users are searching that data.

Performance questionnaire

This questionnaire begins with a single-instance Splunk Enterprise deployment based on the reference architecture described in the Reference machine for single-instance deployments topic. These guidelines help you decide when to distribute your Splunk platform deployment.

Question 1: Do you need to index more than 2GB of data per day?

Question 2: Do you need more than two users signed in at one time?

If you answer **No** to questions 1 and 2, then your Splunk platform instance can share a reference machine for distributed deployments with other Splunk platform services.

If you answer **Yes** to question 1 or 2, then proceed to Question 3.

Note When deploying Splunk Enterprise on Windows OS, do not utilize a host that provides Active Directory or Exchange services, or runs machine virtualization software. Those services are I/O intensive and can reduce Splunk Enterprise indexing and search performance.

Question 3: Do you need to index more than 300GB per day?

Question 4: Do you need more than four concurrent users?

If you answer **No** to questions 3 and 4, then a single dedicated Splunk Enterprise instance running on a reference machine can provide sufficient resources for the indexing and search workload. Go to Question 5.

If you answer **Yes** to question 3 or 4, then scale your Splunk Enterprise deployment to multiple machines to handle the increased demand of indexing and searching. Go to Question 5.

Question 5: Do you need more than 600GB of total storage?

See How Splunk Enterprise calculates disk storage.

If you answer **No**, then a single dedicated reference machine should be able to handle indexing and search workload, but you can consider adding additional storage to the machine to account for increased disk usage due to higher retention. Go to Question 6.

If you answer **Yes**, then scale your Splunk Enterprise deployment to multiple machines to handle the increased demand of indexing and searching. Go to Question 6.

Question 6: Do you want to create or run a Splunk app, alert, or solution that executes more than 8 concurrent saved searches?

Question 7: Do you need to search large quantities of data for a small set (less than 1 per cent) of results?

If you answer **No** to questions 6 and 7, you might not require multiple indexers in your Splunk Enterprise deployment at this time.

If you answer **Yes** to questions 6 or 7, then scale your Splunk Enterprise deployment to multiple machines to handle the increased demand of indexing and searching.

Summary of performance recommendations

The Daily Indexing Volume table summarizes the performance recommendations that were given in the performance checklist. The table shows the number of reference machines that you need to index and search data in Splunk Enterprise, depending on the number of concurrent users and the amounts of data that the instance indexes.

An indexer that meets the minimum reference hardware requirements can ingest up to 300GB/day while supporting a search load. For a review of the current reference hardware specifications, see Reference hardware in this manual.

The table is only a guideline. Modify these figures based on your use case. If you need help defining and scaling a Splunk platform environment, contact your Splunk Sales representative or Professional Services.

Daily Indexing Volume						
	< 2GB/day	2 to 300 GB/day	300 to 600 GB/day	600GB to 1TB/day	1 to 2TB/day	2 to 3TB/day
Total Users: less than 4	1 combined instance	1 combined instance	1 Search Head, 2 Indexers	1 Search Head, 3 Indexers	1 Search Head, 7 Indexers	1 Search Head, 10 Indexers
Total Users: up to 8	1 combined instance	1 Search Head, 1 Indexers	1 Search Head, 2 Indexers	1 Search Head, 3 Indexers	1 Search Head, 8 Indexers	1 Search Head, 12 Indexers
Total Users: up to 16	1 Search Head, 1 Indexers	1 Search Head, 1 Indexers	1 Search Head, 3 Indexers	2 Search Heads, 4 Indexers	2 Search Heads, 10 Indexers	2 Search Heads, 15 Indexers
Total Users: up to 24	1 Search Head, 1 Indexers	1 Search Head, 2 Indexers	2 Search Heads, 3 Indexers	2 Search Heads, 6 Indexers	2 Search Heads, 12 Indexers	3 Search Heads, 18 Indexers
Total Users: up to 48	1 Search Head, 2 Indexers	1 Search Head, 2 Indexers	2 Search Heads, 4 Indexers	2 Search Heads, 7 Indexers	3 Search Heads, 14 Indexers	3 Search Heads, 21 Indexers

Forwarder-to-indexer ratios

Splunk Enterprise indexers are responsible for accepting data streams from internal and external sources, such as forwarders, and indexing that stream locally. Indexing the data requires plentiful disk I/O bandwidth and some computing resources. Indexing capacity remains the top concern when you consider how many forwarders an indexer can handle.

The number of forwarders from which an indexer can accept data depends on several factors:

- Number of CPU cores on the machine. The number of cores should meet or exceed the reference standard.
- The storage available to the machine should meet or exceed the reference standard.

- Whether the indexer runs Windows or *nix.
- The amount of data to be forwarded to the indexers.
- Whether the indexer also acts as a deployment server.

Forwarder-to-indexer ratio testing for a *nix instance

To provide guidance for the estimated number of forwarders that can connect to a single *nix instance of Splunk Enterprise, a test was setup with:

- A Splunk Enterprise instance with 8 cores and 7GB of RAM and 4 x 420GB disks in RAID 0, running a 64-bit Linux OS.
- A high-speed local area network (LAN) operating at 100Mb/s or faster.
- A pool of universal forwarders sending data that was not pre-processed.

In these circumstances, the instance was able to handle a minimum of 2000 forwarders and regularly handled as many as 5000 forwarders.

Performance was best when the server was configured to accept a high number of Unix file descriptors, typically three to four times the number of forwarders that the indexer could accept.

Note: These numbers are for guidance only. Results vary depending on the configuration of the indexers, forwarders, and network.

Splunk Enterprise service limits

The following are Splunk Enterprise service limits and constraints. These service limits are applicable to all Splunk Enterprise subscriptions. You can use this list as guidance to ensure the best experience. Keep in mind that some limits depend on configuration, system load, performance, and available resources.

Service Limit	Best Practice
Number of buckets in the cluster	Max: 40m Recommended: 25m
Number of concurrent users	Max: 2000 Recommended: 500
Number of peers in a non-clustered distributed search environment	Max: 1000 with bestEffortSearch Recommended: 500
Number of alert suppressions in a search head cluster	Max: 100,000 Recommended: 1000
Number of search-heads in a search head cluster	Max: 25

Service Limit	Best Practice
	Recommended: 10
Number of sites in multisite cluster	Max: 6 Recommended: 3

Parallelization settings

New settings are available in Splunk Enterprise to improve search and indexing performance.

Who can use these settings

The parallelization settings are designed to improve the performance of specific components in Splunk Enterprise. The parallelization features are intended for customers with excess CPU cores and I/O capacity to leverage their hardware for improved performance across the indexing tier. You can use these settings to allocate CPU resources to the most common uses for your Splunk platform environment, tuning the indexers to meet that demand.

Summary of settings

Setting	Description
Batch mode search parallelization	Allows a batch mode search to open additional search pipelines on each indexer, processing multiple buckets simultaneously.
Parallel summarization for data models	Allows the scheduler to run concurrent data model acceleration searches on the indexers.
Parallel summarization for report accelerations	Allows the scheduler to run concurrent report acceleration searches on the indexers.
Index parallelization	Allows concurrent data processing pipelines on indexers and forwarders.

If the indexers in your Splunk platform environment exceed the reference hardware specifications, you may review the use case and increase one parallelization settings up to the maximum recommended value. If your indexers are at or near capacity, changing the parallelization settings can have a negative impact on search and indexing performance. All parallelization settings require a service restart to take effect.

Batch mode search parallelization

Batch mode searches are designed to search and return event data by bucket, instead of by time. By adding more batch search pipelines, multiple buckets are processed simultaneously, speeding the return of search results. Customers leveraging batch mode search parallelization can see a doubling of speed in returning batch mode search results.

Setting name	Default	Maximum recommended value	Impact

Setting name	Default	Maximum recommended value	Impact
batch_search_max_pipeline	1	2	Multiplies the number of search pipelines per batch mode search, per indexer.

Adjusting the batch_search_max_pipeline setting in limits.conf to 2 multiplies the IO, processing, and memory used by batch mode searches on every indexer. A value of 2 provides the best performance increase, with higher values succumbing to diminishing returns. For configuration details, see Configure batch mode search parallelization in the Splunk Enterprise *Knowledge Manager Manual*.

Splunk administrators can use the monitoring console to monitor and track indexer resource use. For more details, see About the monitoring console in *Monitoring Splunk Enterprise*.

Parallel summarization

There are two types of accelerated searches: data model accelerations and report accelerations. Both acceleration types create search results on disk beside each index bucket. When a scheduled acceleration search is unable to keep up with the data volume in an index, latency is introduced into the search results. By allowing the scheduler to run concurrent acceleration searches on the indexers, multiple buckets are processed simultaneously, speeding the creation of accelerated search results. Customers leveraging parallel summarization can see a doubling of speed in building accelerated search results.

Data model accelerations

Setting name	Default	Maximum recommended value	Impact
acceleration.max_concurrent	3	3	Multiplies the number of scheduled acceleration searches per data model, per indexer.

The acceleration.max_concurrent setting in datamodels.conf defaults to 3, multiplying the IO, processing, and memory used while running scheduled acceleration searches on every indexer. A value of 3 provides the best performance increase, with higher values succumbing to diminishing returns. For configuration details, see Parallel Summarization in the Splunk Enterprise *Knowledge Manager Manual*

Report accelerations

Setting name	Default	Maximum recommended value	Impact
auto_summarize.max_concurrent	1	2	Multiplies the number of scheduled acceleration searches per search, per indexer.

Adjusting the `auto_summarize.max_concurrent` setting in `savedsearches.conf` to 2 multiplies the IO, processing, and memory used while running scheduled acceleration searches on every indexer. A value of 2 provides the best performance increase, with higher values succumbing to diminishing returns. For configuration details, see Use parallel summarization to speed up creation and maintenance of report summaries in the Splunk Enterprise *Knowledge Manager Manual*.

Splunk administrators can use the monitoring console to monitor and track indexer resource use. For more details, see About the monitoring console in *Monitoring Splunk Enterprise*.

Index parallelization

Index parallelization allows an indexer to maintain multiple pipeline sets. A pipeline set handles the processing of data, from receiving streams of events, through event processing, and writing the events to disk. By allowing an indexer to create and operate multiple pipelines, multiple data streams can be processed with additional CPU cores, accelerating data parsing and disk writing up to the limits of the indexer's I/O capacity. Customers leveraging index parallelization can see an increase in an indexer's sustained indexing load, or a doubling of indexing speed when receiving a sudden surge of data from the forwarders.

Setting name	Default	Maximum recommended value	Impact
`parallelIngestionPipelines`	1	2	Multiplies the number of pipelines per indexer.

Adjusting the `parallelIngestionPipelines` setting in `server.conf` to 2 will use an additional 4-6 CPU cores, and requires 300-400 IOPS to maintain indexing thruput on every indexer. Also, there are fewer CPU cores available for search processing. A value of 2 provides the best performance increase, with higher values succumbing to diminishing returns. For configuration details, see Manage pipeline sets for index parallelization in the Splunk Enterprise *Managing Indexers and Clusters of Indexers Manual*

Splunk administrators can use the monitoring console to monitor and track indexer resource use. For more details, see About the monitoring console in *Monitoring Splunk Enterprise*.