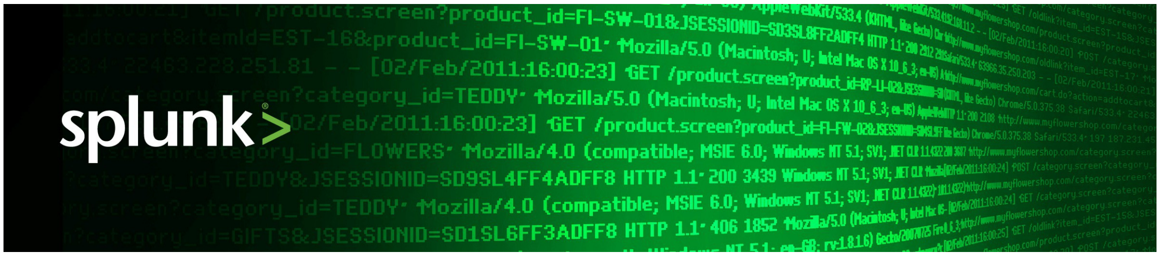


# Splunk® Enterprise Inherit a Splunk Enterprise Deployment 7.2.4

Generated: 11/16/2019 11:32 pm

Copyright (c) 2019 Splunk Inc. All Rights Reserved

# Table of Contents

# Inherited deployment tasks

## Inherited deployments

If you are a system administrator who has inherited the responsibility for a Splunk software deployment, use this manual to gain an understanding of your deployment's network characteristics, data sources, user population, and knowledge objects. This information will help orient you to the essential aspects of the Splunk platform running in your environment. It includes specific suggestions for how to discover what is running, how well it is running, who is using it, and where to go for more detailed information.

For a high-level introduction to Splunk Enterprise software, see the Splunk Enterprise Overview manual.

To learn about the basics of searching and reporting with Splunk software, use the Search Tutorial.

If Splunk software is new to you, there are resources available to help you:

- Splunk Answers
- Splunk Education
- Splunk user groups
- Splunk user groups Slack

The Splunk Professional Services team is also available to perform a technical assessment of your Splunk environment to ensure that your deployment and internal processes follow best practices.

## Draw a diagram of your deployment

Drawing a diagram of your deployment is a useful tool for you to visualize the details as you learn about your deployment, as well as to refer to in the future.

As you read the topics in this manual, create a diagram of your Splunk deployment. Add details as you discover them.

Your diagram can be on paper, which is preferable at the beginning. If you prefer to work with a diagramming tool like Visio or Omnigraffle, here are some icons you can use:

http://wiki.splunk.com/Community:Splunk_Visio_Stencil

Your diagram should show the following items:

- Each search head.
- Each indexer.
- Any additional components such as
    - cluster master (if you have indexer clustering)
    - search head deployer (if you have search head clustering)
    - deployment server
    - license master
    - monitoring console
    - KV store
- Forwarders, or with a large number of forwarders, server classes, which are sets of forwarders.
- The connections between each instance.

Leave room around each node in the diagram so that you can add information as you discover it. For each node, include the following information:

- The version of Splunk Enterprise it is running.
- Whether it is running a KV store.
- All open ports.
- Machine information like operating system, CPU, physical memory, storage type, and virtualization.

See the next topic, Deployment topologies, for definitions of most of the components. Continue to the following topics for steps for discovering them using either the monitoring console, if you have it, or configuration file inspection if you do not have the monitoring console. See Review your apps and add-ons for information and steps for discovering server classes and the KV store in your deployment.

# Deployment topologies

To support a department-sized environment, you might need only a single Splunk Enterprise instance, running on a single machine.

To support larger environments, however, where data originates on many machines and where many users need to search the data, you can scale the deployment by distributing multiple Splunk Enterprise instances across multiple machines, each instance configured to perform a specialized task.

The purpose of this topic, and the topics that immediately follow it, is to help you to determine what role each of the Splunk Enterprise instances in your current deployment performs. If you already have that information, or if your deployment consists of just a single instance, you can skip these topics.

This topic provides an overview of Splunk Enterprise deployments, with a description of the types of topologies and components that a deployment can include. It then outlines procedures that you can use to discover the specifics of your inherited deployment.

## Intended audience

The topology discovery processes, described in this topic and the topics that follow it, are intended for system administrators with little or no Splunk Enterprise experience.

The path to discovery, in its most basic form, requires only a few simple system tools, such as a file browser and a text editor.

There is also an alternative discovery process that uses the Splunk Enterprise monitoring console. The monitoring console provides a graphical overview of your deployment and is readily usable by new Splunk Enterprise administrators. However, its use as a discovery tool requires that the previous Splunk Enterprise administrator already configured it.

Experienced Splunk Enterprise administrators might prefer to use various Splunk-specific tools and methods, such as CLI commands, searches, inspection of log files, and so on, to perform the discovery process more quickly. These methods all require some prior experience with Splunk Enterprise, so they are not immediately usable by the new Splunk Enterprise administrator.

## How Splunk Enterprise scales

The material presented in this topic provides an overview of common Splunk Enterprise deployment topologies and the types of instances that compose them. For more detail, read the *Distributed Deployment Manual*.

Splunk Enterprise performs a number of functions as it processes data. These functions fall into these categories:

**1.** It ingests data from files, the network, or other sources.

**2.** It parses, indexes, and stores the data.

**3.** It runs searches on the indexed data.

To scale your system, you can distribute these functions across multiple specialized instances of Splunk Enterprise. These instances can range in number from just a few to many thousands, depending on the quantity of data, the number of users accessing the data, and other variables in your environment.

For example, your deployment might consist of hundreds of instances that only ingest data, several other instances that index and store the data, and a single instance that manages searches on the data.

## Splunk Enterprise components

A Splunk Enterprise **component** is a Splunk Enterprise instance that performs a specialized task, such as indexing data. There are several types of components, to match the types of tasks in a deployment.

Components fall into two broad categories:

- Processing components. These components handle the data.
- Management components. These components support the activities of the processing components.

### ***Processing components***

The types of processing components are:

- Forwarders
- Indexers
- Search heads

**Forwarders** ingest raw data and forward the data to another component, either another forwarder or an indexer.

Forwarders are usually co-located on machines running applications that generate data, such as web servers.

The **universal forwarder** is the most common type of forwarder. Your deployment might also contain **heavy forwarders** and **light forwarders**.

Usually, forwarders ingest data and forward that data directly to indexers. In some topologies, however, groups of forwarders forward their data to intermediate forwarders, which then forward the consolidated data to indexers. Any type of forwarder can serve as an intermediate forwarder.

**Indexers** index and store data. They also search across the data.

Indexers usually reside on dedicated machines.

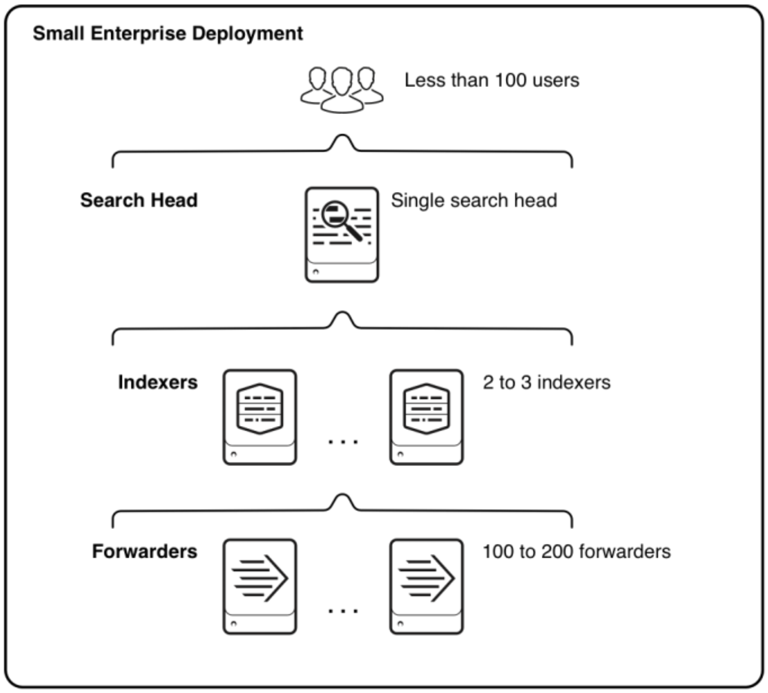
Indexers can be either independent (non-clustered) indexers or clustered indexers. Clustered indexers, also known as **peer nodes**, are nodes in an **indexer cluster**.

**Search heads** manage searches. They handle search requests from users and distribute the requests across the set of indexers, which search their local data. The search head then consolidates the results from all of the indexers and serves them to the users. The search head provides the user with various tools, such as **dashboards**, to assist the search experience.

Search heads usually reside on dedicated machines.

Search heads can be independent search heads, **search head cluster members**, search head nodes in an indexer cluster, or **search head pool members**.

The following diagram of a non-clustered distributed search topology provides a simple example of how the processing components work together to process data. It illustrates the type of deployment that might support the needs of a small enterprise.



The diagram shows the components that support the three main tiers of processing. Starting from the bottom of the diagram, these are the processing tiers:

- **Data input.** Data enters the system through forwarders, which ingest external data, perform a small amount of preprocessing on it, and then forward the data to the indexers. Depending on your data sources, you might have hundreds of forwarders ingesting data.

- **Indexing.** Two or three indexers receive, index, and store incoming data from the forwarders. The indexers also search that data, in response to requests from the search head.

- **Search management.** A single search head manages searches and interacts with users.

To scale the system, you can add more components to each tier. For ease of management, or to meet high availability requirements, you can group components into indexer clusters or search head clusters.

***Management components***

Management components are specially configured versions of Splunk Enterprise instances that support the activities of the processing components. A deployment usually includes one or more of these management components:

- The **monitoring console**, available in Splunk Enterprise 6.2 and later, performs centralized monitoring of the entire deployment. See Use the monitoring console to determine your topology.
- The **deployment server** distributes configuration updates and apps to some processing components, primarily forwarders.
- The **license master** handles Splunk Enterprise licensing.
- The **indexer cluster master** coordinates the activities of an **indexer cluster**. It also handles updates for the cluster.
- The **search head cluster deployer** handles updates for a **search head cluster**.

Your deployment might include all or none of these components, depending on the scale and specifics of your deployment topology.

Multiple management components sometimes share a single Splunk Enterprise instance, perhaps along with a processing component. In large-scale deployments, however, each management component might reside on a dedicated instance.

## Common deployment topologies

The **distributed search** topology provides a flexible way to scale your deployment. Distributed search has many variants, so that your deployment can fit the needs of your organization.

All Splunk Enterprise deployment topologies are variants on distributed search. The variants relate to whether the topology incorporates indexer clustering, search head clustering, or both. In all distributed topologies, forwarders handle data input.

- **Basic distributed search.** In basic distributed search, independent search heads manage searches for a group of independent indexers. See Basic distributed search.

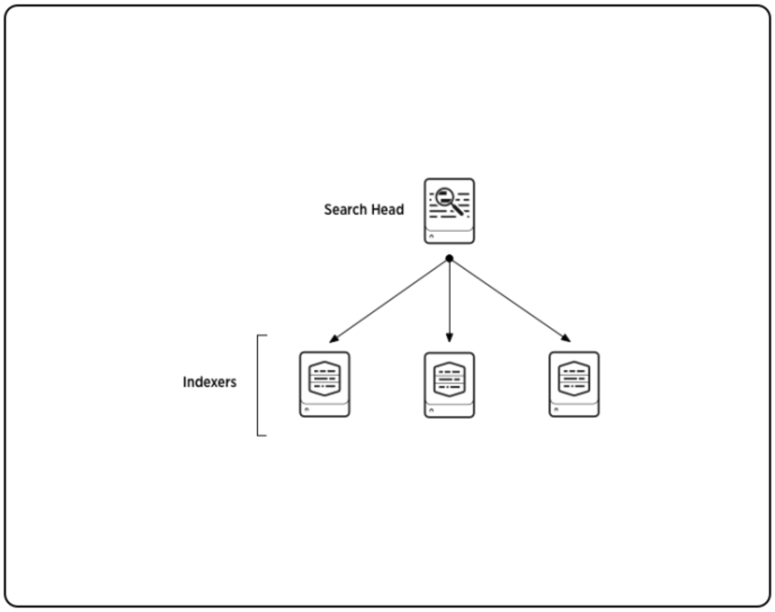
- **Indexer cluster.** In an **indexer cluster**, a group of indexers replicate data among themselves to ensure high data availability. A master node provides centralized management of the indexers. As in basic distributed search, forwarders and search heads handle data input and search management. See Indexer cluster.
- **Search head cluster.** In a **search head cluster**, a group of search heads share search management responsibilities. They distribute searches to indexers, either independent indexers or nodes in an indexer cluster. See Search head cluster.
- **Combined indexer cluster and search head cluster.** This topology is common in larger deployments. It follows the pattern of an indexer cluster, except that the search management function is handled by a search head cluster instead of individual search heads. See Combined indexer cluster and search head cluster.

The *Distributed Deployment Manual* provides extensive descriptions and examples of the full range of deployment topologies.

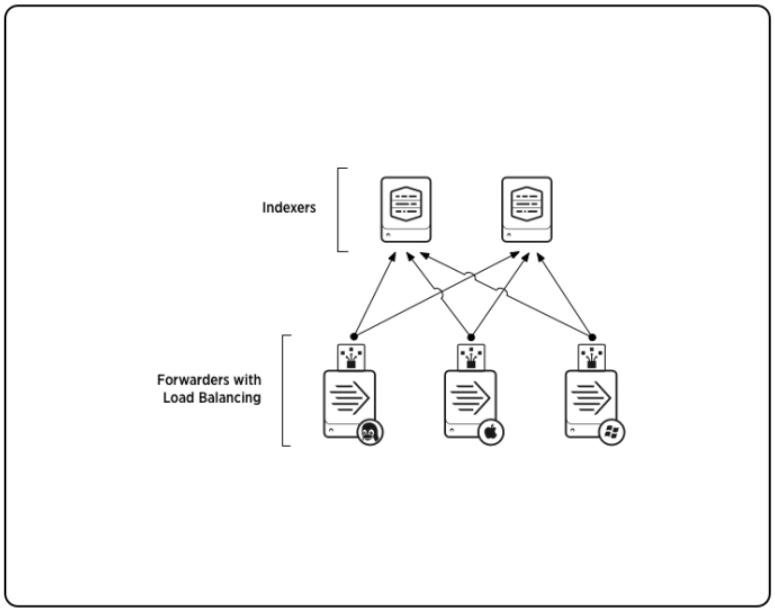
**Note:** There is one other distinct topology, **search head pooling**. In this topology, search heads use shared storage for configuration and user data. This topology is uncommon and has been deprecated in favor of search head clustering, but you might find that your inherited deployment uses search head pooling.

### *Basic distributed search*

This diagram shows a simple distributed search topology, with one search head and three indexers:



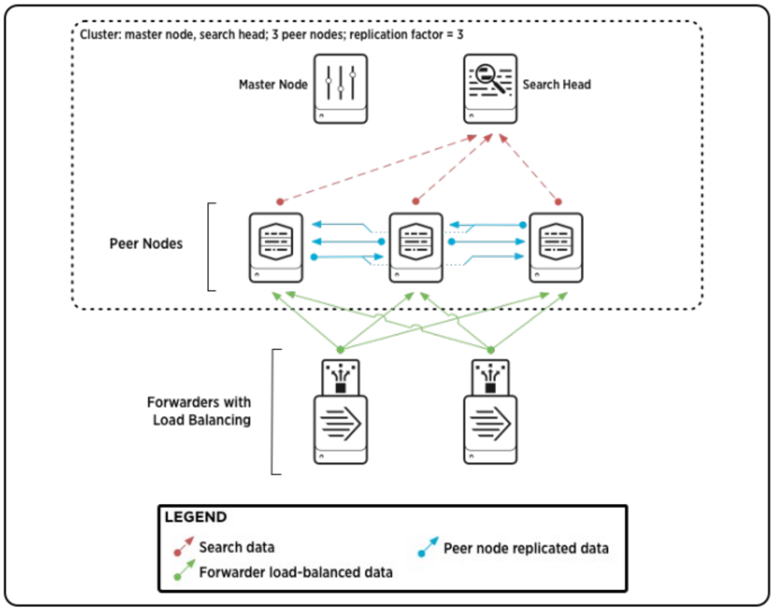
The diagram does not show the forwarders, which ingest the external data and send it to the indexers. Here is a diagram of forwarders employing load balancing to send data to multiple indexers:



For details on distributed search, see About distributed search and the topics that follow it, in *Distributed Search*.

### *Indexer cluster*

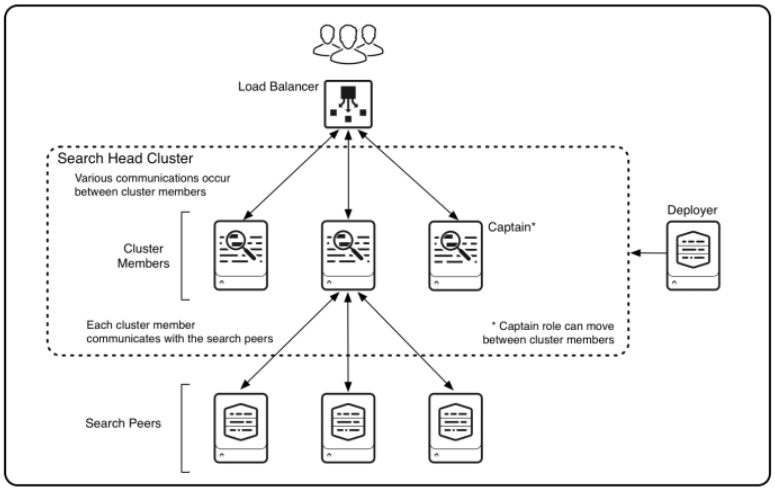
This diagram shows a simple indexer cluster, with one search head and three indexers (peer nodes). A master node controls the interactions among nodes. As in all distributed topologies, forwarders send data to the indexers.


Cluster: master node, search head; 3 peer nodes; replication factor = 3
Master Node
Search Head
Peer Nodes
Forwarders with Load Balancing
LEGEND
Search data
Forwarder load-balanced data
Peer node replicated data

For details on indexer clusters, see About indexer clusters and index replication in *Managing Indexers and Clusters of Indexers*.

### *Search head cluster*

This diagram shows a simple search head cluster, with three search heads, or "members." The search heads coordinate with three independent indexers, or "search peers."


Load Balancer
Search Head Cluster
Various communications occur between cluster members
Cluster Members
Captain*
Deployer
Each cluster member communicates with the search peers
* Captain role can move between cluster members
Search Peers

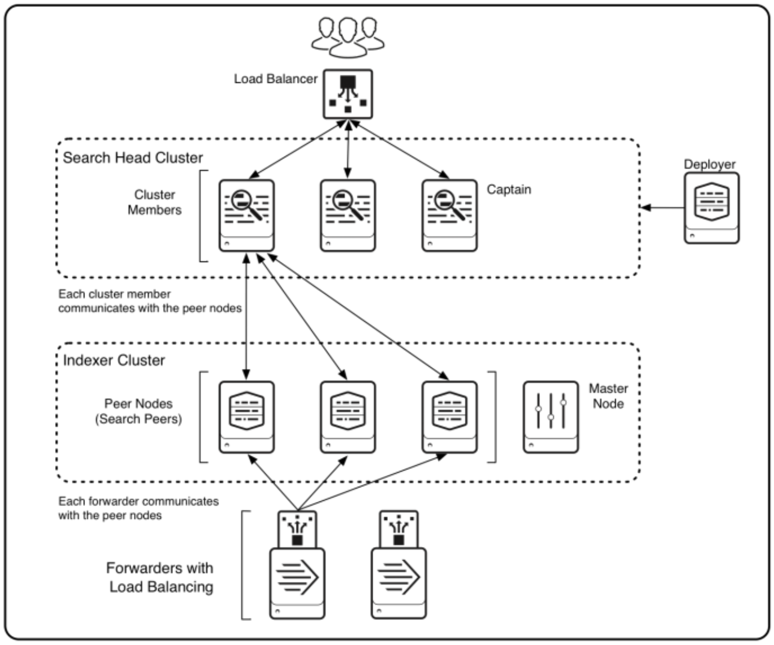
As in all distributed topologies, forwarders (not shown) ingest the external data and send it to the indexers.

A **deployer** resides outside the search head cluster and handles certain updates to cluster configurations.

For details on search head clusters, see About search head clustering and the topics that follow it, in *Distributed Search*.

### *Combined indexer cluster and search head cluster*

In this diagram, a search head cluster manages searches across an indexer cluster:


Load Balancer
Search Head Cluster
Cluster Members
Captain
Deployer
Each cluster member communicates with the peer nodes
Indexer Cluster
Peer Nodes (Search Peers)
Master Node
Each forwarder communicates with the peer nodes
Forwarders with Load Balancing

For details on combining an indexer cluster with a search head cluster, see Integrate the search head cluster with an indexer cluster in *Distributed Search*.

## Path to discovery

To determine your deployment topology, you must identify the components and their relationships.

Discovery involves these steps:

1. Locate your Splunk Enterprise and universal forwarder instances.

   Determine which machines contain instances of your deployment. Although it is possible for a single machine to host multiple instances, such a configuration is unusual except in test environments. In production environments, each Splunk Enterprise instance usually resides on its own machine.
2. Identify your components.

   For each instance, identify the components that it hosts. Components define the roles that the instances play in the deployment. A single instance can host multiple components.
3. Identify the relationships between components.

   Determine how the components participate in the overall deployment topology.

It can be helpful to draw a diagram of your deployment, as you go about the discovery process. See Draw a diagram of your deployment.

### *1. Locate your Splunk Enterprise and universal forwarder instances*

The first step is to locate the Splunk Enterprise and universal forwarder instances on your machines. Note these points:

- All components run on Splunk Enterprise instances, except for the universal forwarder. The universal forwarder is a lightweight version of Splunk Enterprise with its own executable.
- Splunk Enterprise instances usually reside on dedicated machines, as a best practice. However, you might discover an instance running on a machine that is also performing some entirely different function.
- Universal forwarder instances usually reside on machines that host other applications, such as web servers. The forwarders ingest data produced by those applications.
- A single machine can host multiple instances, although the best practice is for each instance to reside on its own machine.
- The absence of **Splunk Web**, the Splunk graphical user interface, is not a reliable indicator that the machine does not host a Splunk Enterprise instance. On most deployments, only a subset of Splunk Enterprise instances, such as search heads and some management components, have a running web interface.

You can identify machines hosting Splunk Enterprise and universal forwarder instances by looking for the presence of Splunk subdirectories on the machines' file systems.

Splunk documentation refers to the base directory for the Splunk file system as `$SPLUNK_HOME`.

Instances typically reside under these locations on a file system:

| Operating system | Locations for Splunk Enterprise $SPLUNK_HOME | Locations for universal forwarder $SPLUNK_HOME |
|---|---|---|
| Windows | `\Program Files\Splunk` | `\Program Files\SplunkUniversalForwarder` |

| Operating system | Locations for Splunk Enterprise $SPLUNK_HOME | Locations for universal forwarder $SPLUNK_HOME |
|---|---|---|
|  | `C:\Splunk`<br>`C:\SPL` | `C:\SplunkUniversalForwarder` |
| Linux<br>Solaris<br>AIX<br>HP-UX<br>FreeBSD | `/opt/splunk` | `/opt/splunkforwarder` |
| Mac OS X | `/Applications/splunk` | `/Applications/splunkforwarder` |

**Caution:** This table shows default or typical locations for `$SPLUNK_HOME`. However, the installation process permits the user to install to any location and to change the name of the base directory. Therefore, if you cannot immediately identify `$SPLUNK_HOME`, look for a directory that contains a set of Splunk subdirectories. These subdirectories include `bin, etc, include, lib, openssl, share`, and `var`.

You can also verify that the machine hosts `$SPLUNK_HOME` by looking for a `bin` subdirectory that contains the `splunk, splunkd`, and `btool` executables, among others. The parent of that `bin` subdirectory is `$SPLUNK_HOME`.

Once you identify a machine with an installed instance, confirm that the instance is currently running. Use a system tool such as `ps` or Task Manager to look for the `splunkd` process.

### *2. Identify your components*

You can identify your components with either of these methods:

- Use the monitoring console.
- Examine each instance's configuration files.

If your Splunk Enterprise deployment has a monitoring console running, use it to discover the components and their relationships. See Use the monitoring console to determine your topology.

If your Splunk Enterprise deployment does not have a monitoring console, you must examine each instance's configurations. Browse its set of configuration files, which are text files that hold all of the instance's configurations. See

Examine configuration files to determine your topology.

See Splunk Enterprise components.

### *3. Identify the relationships between components*

When you know the components, the relationships between them are usually apparent. For example, if you have a search head and three indexers in a non-clustered environment, each indexer is a **search peer** of the search head, meaning that the indexer processes search requests for the search head. Similarly, if you find that you have components of an indexer cluster, then your deployment contains an indexer cluster.

If your deployment has a monitoring console, you can use it to identify the relationships, as well as the components themselves.

Your deployment topology will usually fall into one of these broad categories:

- Basic distributed search
- Indexer cluster
- Search head cluster
- Combined indexer cluster and search head cluster

See Common deployment topologies.

### *Summary of component types*

This summary outlines the main points to keep in mind as you perform the component discovery process.

A Splunk Enterprise deployment consists of instances that function as processing and management components. A deployment usually contains only a subset of possible component types. In the discovery process, you identify the components that reside on each instance.

An instance ordinarily hosts at most a single processing component, although a processing component can also perform a secondary processing function. For example, some search heads forwards their internal data to indexers. The forwarding function on a search head is strictly secondary to its main function, however, as the forwarding involves internal data only.

Management components are frequently co-located on an instance with a processing component or other management components.

Some of the processing component types have variants. For example, an indexer can be independent or a peer node of an indexer cluster.

These are the processing components and their variants:

- Search head, which can be any of these types:
    - Independent search head
    - A search head node of an indexer cluster
    - A member of a search head cluster
    - A search head node of an indexer cluster and a member of a search head cluster
    - A member of a search head pool
- Indexer, which can be any of these types:
    - Independent indexer
    - A peer node of an indexer cluster
- Forwarder, which can be any of these types:
    - Universal forwarders
    - Heavy forwarders
    - Light forwarders
    - Intermediate forwarders (secondary characteristic for any type of forwarder)

These are the management components:

- Monitoring console
- Deployment server
- License master
- Indexer cluster master
- Search head cluster deployer

# Use the monitoring console to determine your topology

If your outgoing admin left you information about a monitoring console (formerly called a distributed management console, DMC), you can use this to discover your deployment's topology.

## Prerequisites

Read Deployment topologies. This topic describes the elements of Splunk Enterprise deployments and offers essential guidance on how to discover your deployment topology.

## Access the monitoring console

Refer to any information provided by your previous administrator or by your organization.

The monitoring console can be hosted on its own instance, or it can be colocated with a cluster master. Less commonly, it can be colocated with another management component. See Which instance should host the console? in *Monitoring Splunk Enterprise*.

Log into Splunk Web on a node that is likely to be the monitoring console. If you do not know what node is your monitoring console, try a cluster master. If that does not give you results, try any search head.

When you have found the node that runs the monitoring console, navigate to the monitoring console:

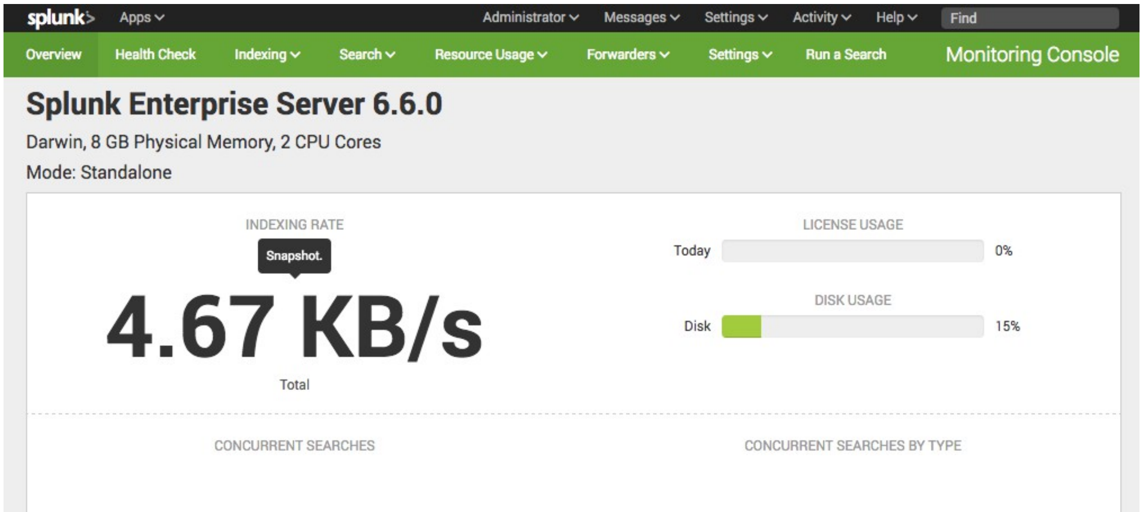
1. Click **Settings** from anywhere in Splunk Web
2. Click the Monitoring Console icon on the left of the panel to open the monitoring console.

## Monitoring console overview

The home page of the monitoring console is the Overview page.

The monitoring console has two modes, standalone and distributed. On the basis of the following screen shots, determine whether the monitoring console on your instance is configured in standalone or distributed mode.

The overview in standalone mode:



Splunk Enterprise Server 6.6.0

Darwin, 8 GB Physical Memory, 2 CPU Cores

Mode: Standalone

INDEXING RATE

4.67 KB/s

Total

LICENSE USAGE

Today 0%

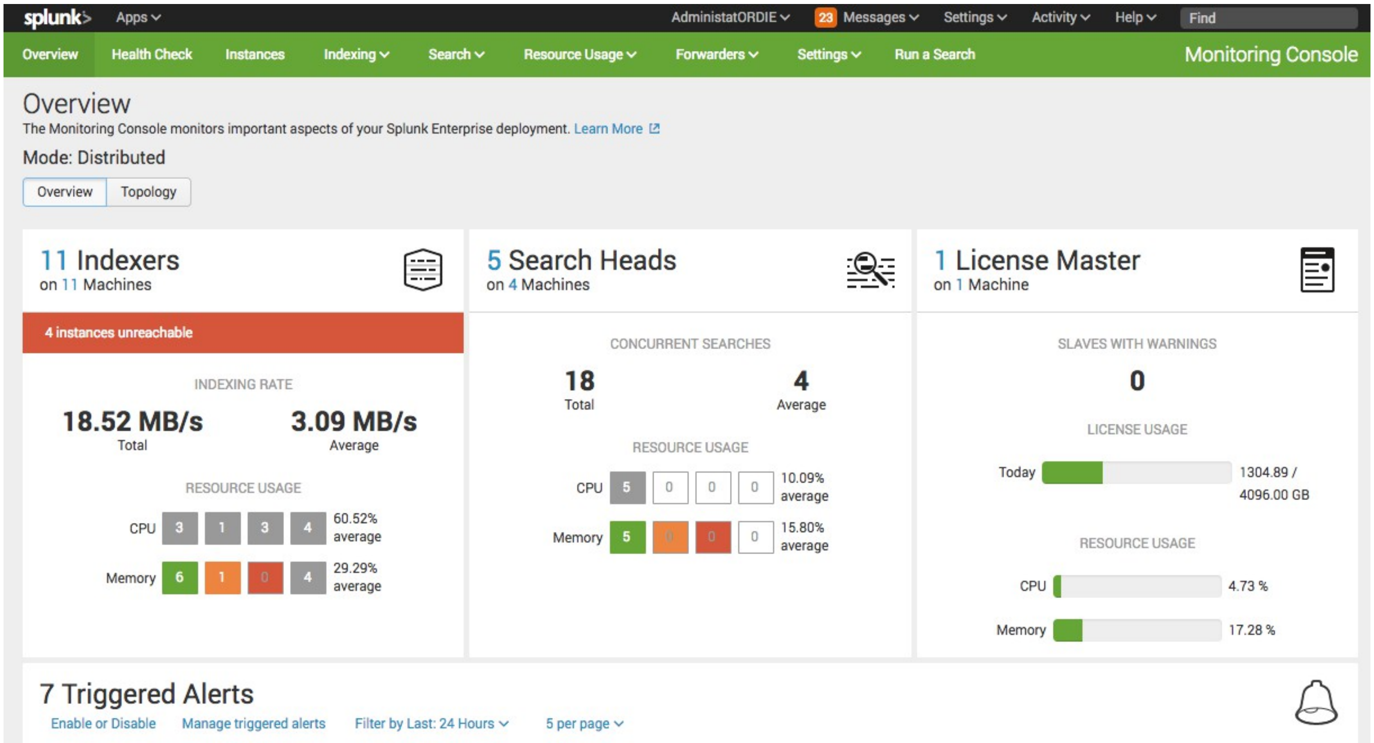
DISK USAGE

Disk 15%

CONCURRENT SEARCHES

CONCURRENT SEARCHES BY TYPE

The overview in distributed mode:



Overview

The Monitoring Console monitors important aspects of your Splunk Enterprise deployment. Learn More

Mode: Distributed

11 Indexers

on 11 Machines

4 instances unreachable

INDEXING RATE

18.52 MB/s Total

3.09 MB/s Average

RESOURCE USAGE

| | | | | | |
|---|---|---|---|---|---|
| CPU | 3 | 1 | 3 | 4 | 60.52% average |
| Memory | 6 | 1 | 0 | 4 | 29.29% average |

5 Search Heads

on 4 Machines

CONCURRENT SEARCHES

18 Total

4 Average

RESOURCE USAGE

| | | | | | |
|---|---|---|---|---|---|
| CPU | 5 | 0 | 0 | 0 | 10.09% average |
| Memory | 5 | 0 | 0 | 0 | 15.80% average |

1 License Master

on 1 Machine

SLAVES WITH WARNINGS

0

LICENSE USAGE

Today 1304.89 / 4096.00 GB

RESOURCE USAGE

CPU 4.73 %

Memory 17.28 %

7 Triggered Alerts

**If you do not have the monitoring console configured in distributed mode on any instance in your deployment, do not set it up at this point**. Instead, continue to Examine configuration files to determine your topology.

## Discover components and topology from the monitoring console

From the Overview page, click the **Topology** toggle to learn about your deployment's topology.

1. See the instances in your deployment.
2. Click each instance to view details including its Splunk Enterprise version.
3. Record this information on your deployment diagram.

See all of the components colocated on an instance:

1. Click **Instances**.
2. Use the **Group** dropdown to view each component in your deployment, leaving KV store for the last.
3. For each instance, note the information displayed in the table. The information under "Role" is the component.
4. Record this information on your deployment diagram.

An admin can optionally set up forwarder monitoring in the monitoring console, as of Splunk Enterprise 6.3.0. To view forwarder information:

1. Click **Forwarders > Forwarder deployment**.
2. Use the **Split by** dropdown to understand your set of forwarders.
3. Scroll down to the Status and Configuration panel.
4. Record information about forwarder type, Splunk version, OS, and system architecture on your diagram.

Do not yet enable forwarder monitoring.

You can optionally rebuild the forwarder asset table. If a forwarder is decommissioned, it remains on the forwarder dashboards until you rebuild the forwarder asset table. This step is probably not immediately necessary, but if you find that your forwarder dashboards contain null results from several forwarders, you might want to rebuild the asset table. If you have many forwarders, it can take a while to run.

1. In the monitoring console, click **Settings > Forwarder Monitoring Setup**.
2. Click **Rebuild forwarder assets**.
3. Select a time range or leave the default of 24 hours.
4. Click **Start Rebuild**.

You will survey groups of forwarders called server classes later, in Review your apps and add-ons.

### Validate your monitoring console setup

If you have used the monitoring console to populate your diagram, it is almost complete.

To ensure accuracy, validate that the monitoring console was correctly configured by your previous administrator. Use the configuration file methods that follow. To validate the monitoring console setup:

1. Test one or two of the instances with multiple server roles, also known as components.
2. Verify that the server roles displayed for that instance on the monitoring console **Instances** page matches the information you gather using the configuration file method.
3. If it does not, investigate configuration files of other instances and populate your deployment diagram with that information.
4. After you complete the rest of the orientation tasks in this manual and reach Monitor system health, correct the monitoring console setup.

## Examine configuration files to determine your topology

In this method of discovery, you examine certain configuration files residing on each Splunk Enterprise instance. The files contain settings whose presence or absence help you to determine what **component** the instance functions as. The settings also help determine the relationships between components, and thus, the overall topology.

The discovery process looks for the characteristic configurations of each component type.

It can be helpful to draw a diagram of your deployment, as you go about the discovery process. See Draw a diagram of your deployment.

## Prerequisites

Read the following material:

- The chapter Administer Splunk Enterprise with configuration files in the *Admin Manual.* The topics in this chapter provide important background information on configuration files: what they are, where they reside, and how they layer on top of one another.

- Deployment topologies. This topic describes the elements of Splunk Enterprise deployments and offers essential guidance on how to discover your deployment topology.

## Configuration file locations

Copies of configuration files can reside in several locations, including system directories and app directories. If a configuration file has copies in multiple locations, the operative value of each setting in the file is determined by a process of file layering, based on an order of precedence.

For details on configuration file locations, see Configuration file directories in the *Admin Manual.* To learn how multiple copies of a configuration file layer, see Configuration file precedence in the *Admin Manual.*

The files that contain component configurations, such as `server.conf`, usually reside only under `$SPLUNK_HOME/etc/system/local`, because component settings are system-level configurations and should not be app-dependent. To be certain of your configurations, however, you must examine all possible locations for copies of each relevant file.

## How to examine configuration files

To examine a configuration file, you can use either a text editor or the `btool` utility.

**Caution:** If you use a text editor, do not make any changes to the files.

The `btool` utility in `$SPLUNK_HOME/bin` provides a quick way to sift through all copies of a configuration file. The advantage of `btool` is that it reports back the final set of configurations that results after layering all copies of the file. For syntax and other details, see Use btool to troubleshoot configurations in the *Troubleshooting Manual*.

## Discover processing components

First, examine each instance to see if it hosts a processing component. Later, you examine the instances to see if they host management components.

To discover the processing components, follow these two procedures:

1. Identify **search heads** and **indexers**.
2. Identify **forwarders**.

An instance typically contains at most one processing component. Therefore, for each instance, follow each procedure, in order, until you identify what processing component, if any, the instance contains.

Processing components in the procedures are identified by bold text.

## Identify search heads and indexers

Determine whether the instance is a search head or an indexer and, if so, what type of search head or indexer.

1. Examine `server.conf` in `$SPLUNK_HOME/etc/system/local`.
    1. Look for a `[clustering]` stanza. If found, this instance is a node of an **indexer cluster**.

        To determine the indexer cluster node type, examine the `mode` setting:
        1. If `mode = master`, this instance is an **indexer cluster master node.** Stop here.

            **Note:** The master node is a management component, not a processing component. Because it is configured in `server.conf`, however, it makes sense to look for it while examining `server.conf` for processing components.
        2. If `mode = slave`, this instance is an **indexer cluster peer node,** also known as a **clustered indexer**. Stop here.
        3. If `mode = searchhead` , this instance is an **indexer cluster search head node.** Continue to the next step, to determine whether this indexer cluster search head is also a member of

a **search head cluster**.

   2. Look for a `[shclustering]` stanza. If found, this instance is a **search head cluster member.** Stop here.

      **Note:** Depending on the deployment topology, a search head cluster member can also be a search head node in an indexer cluster.
   3. Look for a `[pooling]` stanza. If found, this instance is a **a search head pool member.** Stop here.
2. Examine `distsearch.conf` in `$SPLUNK_HOME/etc/system/local`.
   1. Look for a populated `servers` setting in the `[distributedSearch]` stanza. If found, this instance is an **independent search head.** Stop here.

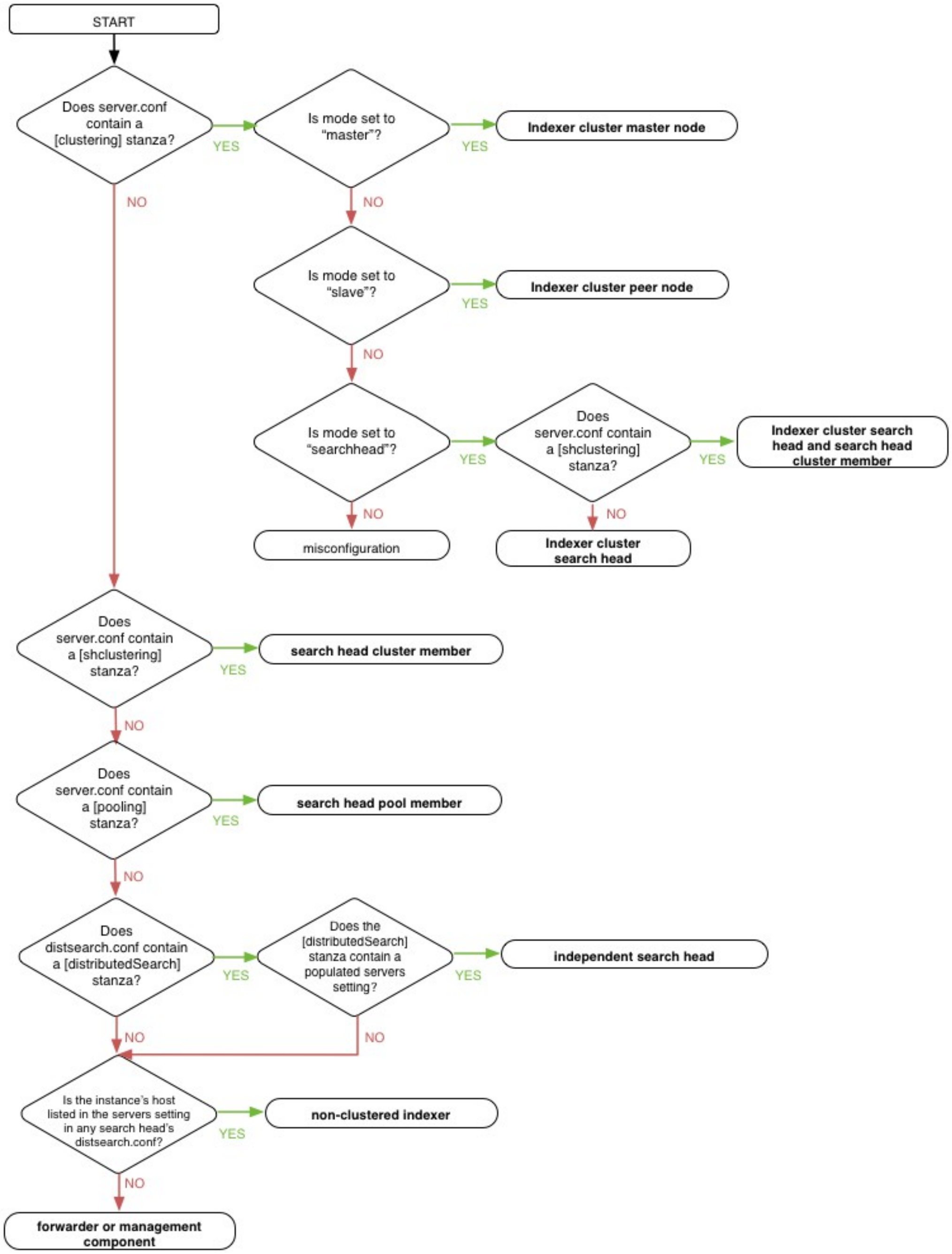
      **Note:** Search heads that are members of search head clusters or search head pools also have a populated `servers` setting. However, you have already identified such search heads in earlier steps of this procedure, so any search heads with a populated `servers` setting that remain to be identified at this point of the procedure must be independent search heads.
3. On all instances previously identified as either search head cluster members, search head pool members, or independent search heads, examine the `servers` setting in `distsearch.conf`.

   The list of addresses in `servers` specifies the hosts for the indexers that this search head connects to. Use this list to determine which instances in your deployment are **independent (non-clustered) indexers.** Stop here.

   **Note:** In an indexer cluster, search heads do not use the `servers` list to specify their indexers. Therefore, if an indexer appears in this list, it is a non-clustered indexer.

Once you have identified your search heads and indexers, any remaining instances are either forwarders or management components.

The following flowchart encapsulates the procedure above:


START
Does server.conf contain a [clustering] stanza?
YES
NO
Is mode set to "master"?
YES
Indexer cluster master node
NO
Is mode set to "slave"?
YES
Indexer cluster peer node
NO
Is mode set to "searchhead"?
YES
Does server.conf contain a [shclustering] stanza?
YES
Indexer cluster search head and search head cluster member
NO
misconfiguration
NO
Indexer cluster search head
Does server.conf contain a [shclustering] stanza?
YES
search head cluster member
NO
Does server.conf contain a [pooling] stanza?
YES
search head pool member
NO
Does distsearch.conf contain a [distributedSearch] stanza?
YES
Does the [distributedSearch] stanza contain a populated servers setting?
YES
independent search head
NO
NO
Is the instance's host listed in the servers setting in any search head's distsearch.conf?
YES
non-clustered indexer
NO
forwarder or management component

**Caution:** In complex deployments, a search head might manage searches for multiple indexer clusters, or for both an indexer cluster and a set of independent indexers. If you suspect that you have such a deployment topology, you can investigate the search head's configuration more deeply. See Search across multiple indexer clusters and Search across both clustered and non-clustered search peers in *Managing Indexers and Clusters of Indexers*.

## Identify forwarders

If the instance is not an indexer or a search head, it might be a forwarder.

The indexers in a Splunk deployment provide information on the forwarders that have been installed in that deployment. Indexers in a deployment keep logs of each forwarder connection and type.

### *Use Splunk search to identify forwarders*

You can get a list of every forwarder that has connected to a Splunk indexer or indexer cluster by logging into Splunk Enterprise on an indexer or search head and running the following search in Splunk Web:

```
index=_internal source=*metrics.log group=tcpin_connections | where
isnotnull(fwdType) | eval sourceHost=if(isnull(hostname),
sourceHost,hostname) | dedup sourceHost | eval connectionType
=case(fwdType=="uf","Universal", fwdType=="lwf", "Lightweight",
fwdType=="full","Heavy") | rename sourceIp as "Source IP", sourceHost
as "Source Host", connectionType as "Forwarder Type" | table "Source IP"
"Source Host" "Forwarder Type"
```

If you have the IP address or host name of a machine whose forwarder status and type you want to check, you can modify the search to specify the host name or IP address:

```
index=_internal source=*metrics.log group=tcpin_connections
[sourceIp=<ip address>|hostname=<host name>] | where isnotnull(fwdType)
| eval sourceHost=if(isnull(hostname), sourceHost,hostname) | dedup
sourceHost | eval connectionType =case(fwdType=="uf","Universal",
fwdType=="lwf", "Lightweight", fwdType=="full", "Heavy") | rename
sourceIp as "Source IP", sourceHost as "Source Host", connectionType as
"Forwarder Type" | table "Source IP" "Source Host" "Forwarder Type"
```

If the table that these searches generate does not list information on the machine you want information about, or you do not have access to Splunk Web, you can examine the machine itself to determine the forwarder type.

### *Manually identify forwarders*

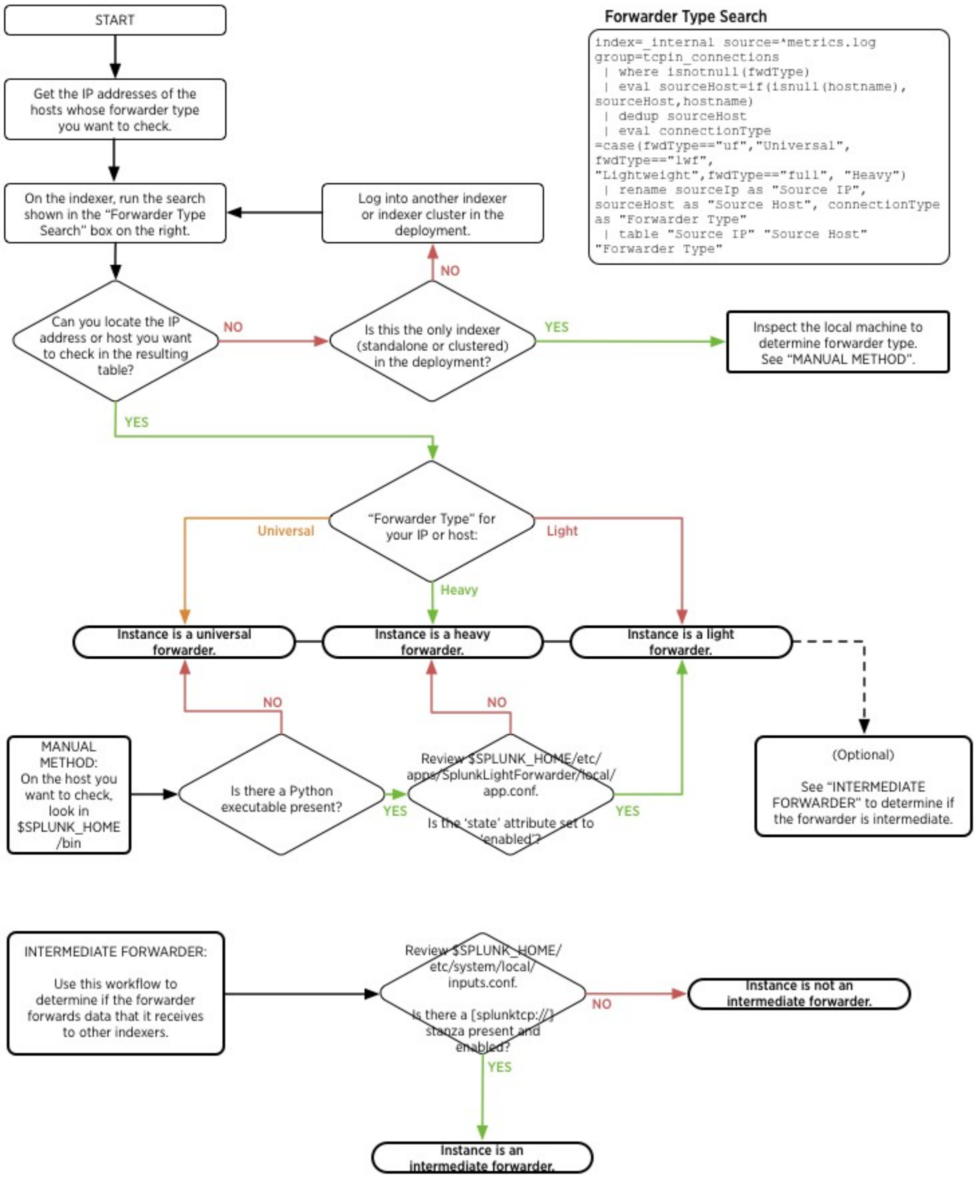
1. On the machine whose status you want to check, look in the `$SPLUNK_HOME/bin` directory. Check to see if a `python` executable (`python.exe` on Windows) is present in this directory.
    1. If there is not, then this instance is a **universal forwarder**. You can stop here or proceed to Identify intermediate forwarders in the next section.
2. If there is, then review `$SPLUNK_HOME/etc/apps/SplunkLightForwarder/local/app.conf` to see if the `state` setting in that file is set to `enabled`.
    1. If it is, then the instance is a **light forwarder.** Proceed to Identify intermediate forwarders in the next section in either case.
    2. If it is not, then the instance is a **heavy forwarder.** Proceed to Identify intermediate forwarders in the next section in either case.

### *Identify intermediate forwarders*

Intermediate forwarders receive data from other forwarders to send to indexers. Any forwarder can be an intermediate forwarder.

1. Review `$SPLUNK_HOME/etc/system/local/inputs.conf` on the forwarder. Is there a `splunktcp://` stanza present and enabled?
    - ♦ If there is, then the instance is an intermediate forwarder.
    - ♦ If there is not, then the instance is not an intermediate forwarder.

See the following flowchart to understand how forwarder discovery works:



START
Get the IP addresses of the hosts whose forwarder type you want to check.
On the indexer, run the search shown in the "Forwarder Type Search" box on the right.
Log into another indexer or indexer cluster in the deployment.
NO
Can you locate the IP address or host you want to check in the resulting table?
NO
Is this the only indexer (standalone or clustered) in the deployment?
YES
Inspect the local machine to determine forwarder type. See "MANUAL METHOD".
YES
"Forwarder Type" for your IP or host:
Universal
Light
Heavy
Instance is a universal forwarder.
Instance is a heavy forwarder.
Instance is a light forwarder.
NO
NO
MANUAL METHOD: On the host you want to check, look in $SPLUNK_HOME /bin
Is there a Python executable present?
YES
Review $SPLUNK_HOME/etc/ apps/SplunkLightForwarder/local/ app.conf.
Is the 'state' attribute set to 'enabled'?
YES
(Optional)
See "INTERMEDIATE FORWARDER" to determine if the forwarder is intermediate.
INTERMEDIATE FORWARDER:
Use this workflow to determine if the forwarder forwards data that it receives to other indexers.
Review $SPLUNK_HOME/ etc/system/local/ inputs.conf.
Is there a [splunktcp://] stanza present and enabled?
NO
Instance is not an intermediate forwarder.
YES
Instance is an intermediate forwarder.

**Forwarder Type Search**

```
index=_internal source=*metrics.log
group=tcpin_connections
 | where isnotnull(fwdType)
 | eval sourceHost=if(isnull(hostname),
sourceHost,hostname)
 | dedup sourceHost
 | eval connectionType
=case(fwdType=="uf","Universal",
fwdType=="lwf",
"Lightweight",fwdType=="full", "Heavy")
 | rename sourceIp as "Source IP",
sourceHost as "Source Host", connectionType
as "Forwarder Type"
 | table "Source IP" "Source Host"
"Forwarder Type"
```

## From processing components to management components

Once you reach this point for all your Splunk Enterprise instances, you know all your processing components, as well as the cluster master management component, if any.

If you cannot find any processing component on a particular instance, the instance hosts management components only.

In the next section, determine which instances host management components.

## Discover management components

Management components can be located on their own dedicated Splunk Enterprise instance, co-located with a processing component, or co-located with other management components. Therefore, any of your Splunk Enterprise instances could contain management components.

Universal forwarder instances, however, cannot also contain management components.

Also, there is usually at most one of each management component for an entire deployment So, for example, once you identify the instance that hosts the deployment server, if any, you can stop looking for other deployment server instances.

This table shows, for each management component, its key indicator, along with the configuration file that pertains to the indicator, if any.

| Component type | Configuration file | Key indicator |
|---|---|---|
| Monitoring console | `$SPLUNK_HOME/etc/apps/splunk_monitoring_console`<br>/splunk_management_console_assets.conf | the existence of<br>`splunk_management_console_` |
| **Deployment server** | `$SPLUNK_HOME/etc/system/local/serverclass.conf` | `[serverClass:<name>]` stanza |
| **License master** | `$SPLUNK_HOME/etc/system/local/server.conf` | `[license]` stanza with `master` |
| **Indexer cluster master** | `$SPLUNK_HOME/etc/system/local/server.conf` | `[cluster]` stanza with `mode =` |

| Component type | Configuration file | Key indicator |
|---|---|---|
| | | |
| **Search head cluster deployer** | N/A | populated `$SPLUNK_HOME/etc/`<br>directory |

If you have completed all the steps in this topic, you should now understand the specifics of your Splunk Enterprise topology. You should know the location and function of each instance and the relationships between instances.

## Components and their relationship with the network

Splunk Enterprise components require network connectivity to work properly if they have been distributed across multiple machines, and even in cases where the components are on one machine.

Splunk components communicate with each other using TCP and UDP network protocols. A firewall that has not been configured to allow these ports open can block communication between the Splunk instances.

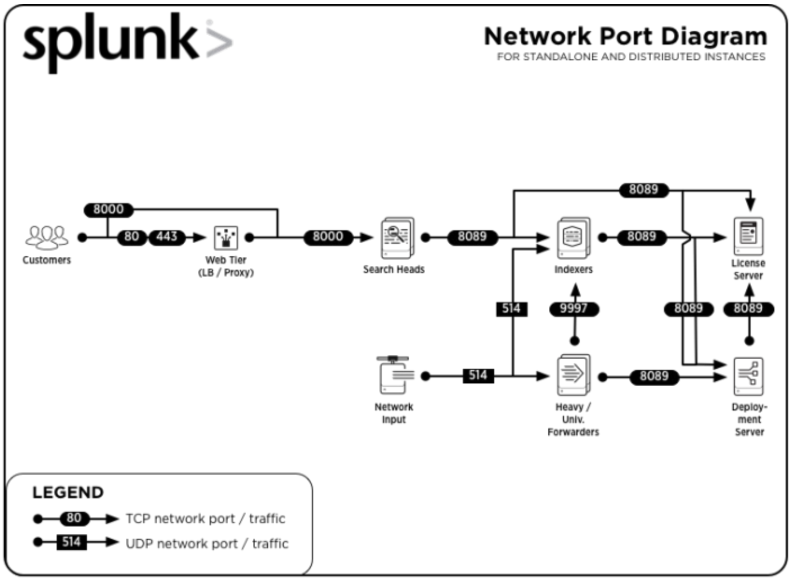
Splunk software uses the following network ports to communicate between its components by default or by convention. You can perform a network port scan on a host to determine if it is listening on a port. Record open port numbers on your deployment diagram.

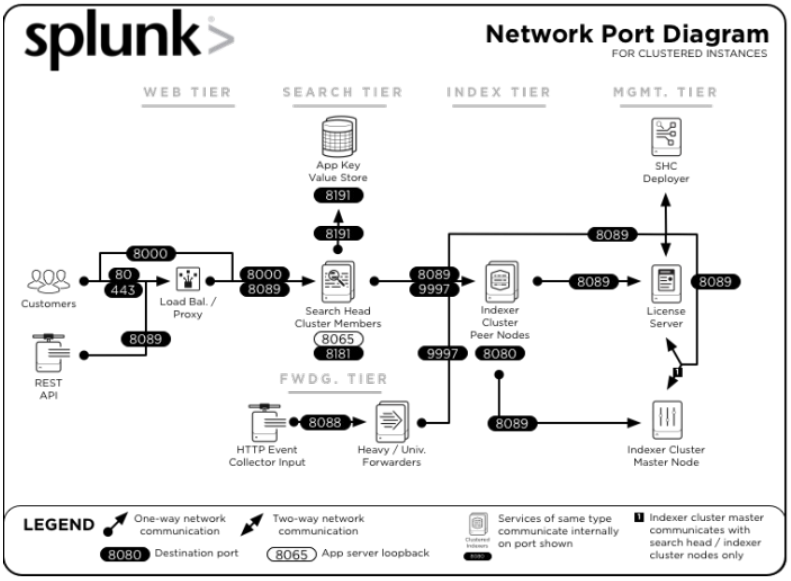
| Component | Purpose | Communicates on | Listens on |
|---|---|---|---|
| All components* | Management / REST API | N/A | TCP/8089 |
| Search head / Indexer | Splunk Web access | Any | TCP/8000 |
| Search head | App Key Value Store | Any | TCP/8065,<br>TCP/8191 |
| Indexer | Receiving data from forwarders | N/A | TCP/9997 |
| Indexer cluster peer node / Search head cluster member | Cluster replication | N/A | TCP/9887 |

| Component | Purpose | Communicates on | Listens on |
|---|---|---|---|
| | | | |
| Indexer/Forwarder | Network input (syslog) | N/A | UDP/514 |

## Diagrams

The following diagrams show the network ports that Splunk software listens on.





# Learn about the data in your Splunk deployment

After you have discovered and diagrammed the topology of your Splunk deployment, the next task is to learn about the data in the deployment.

There are two parts to understand the data in a Splunk deployment. The first part is about how stored data is managed in the deployment. The second part is how the Splunk deployment ingests the data.

## Learn about stored data

Before you assumed control of the deployment, it was configured to ingest data from certain data **sources**. The person or group who owned the data determined the following:

- The amount of available data
- The relevance of the data to the organization
- The length of time that the organization wanted to keep the data that Splunk software ingested
- The data retention policy of the organization
- The people that required access to the data
- The need for any sensitive data to be anonymized

They then worked with other groups to set up Splunk software to get the data indexed and stored.

You can learn about the data that has been indexed by Splunk software with the following methods:

- Review the data summary
- Run searches on the data

### *Review the data summary*

With the Data Summary in Splunk Web, you can determine data sources, **source types**, and the hosts that generated the data. This is the most comprehensive way of learning what data is present in a Splunk deployment.

1. Log into the Splunk instance. If the deployment is distributed, log into a search head.
2. Click **Search and Reporting**.
3. Click **Data Summary**.

4. Click on one of the tabs to get information about the **Hosts**, **Sources**, or **Sourcetypes** that the instance has indexed.
5. (Optional) Click on an entry in the **Data Summary** list to run a search that contains that entry in its results.

For more information about the Search app, see About the Search app in the *Search Manual*.

### *Run searches on the data*

With Splunk **search**, you can create a timeline that shows when the data was ingested by running search commands and adjusting timeline parameters. The kinds of searches you want to run depend on the kind of data you are searching for. You can use the Data Summary to learn what has been indexed into the instance and what you can search for.

1. Log into the Splunk instance.
2. Click **Search and Reporting**.
3. Enter a search that represents the data that you expect to see. If you do not know what data you have, you can use the Data Summary.
4. (Optional) Use the event timeline to determine how far the events go back.
5. (Optional) Set the time picker to a different time range to see events that occur only during that range.
6. Click on individual items in the results to change search parameters or run a new search based on that item.

For information about searching, see Anatomy of a search in the *Search Manual*.

## Learn about the data generators in the deployment

For Splunk software to receive data, it must be configured with **data inputs**. Inputs can be configured on the Splunk indexer, but in most deployments, **forwarders** are configured with the inputs and do the data collection. The data flows from the forwarders into the indexer where Splunk software breaks up the data into **events** that can be form the basis for searches, **reports**, and **dashboards**, or be modified to fit the needs of the data consumers in your organization.

Splunk software can ingest many different kinds of machine data. The *Getting Data In Manual* provides information on the machine data that Splunk software can ingest, and includes but is not limited to:

- Log files

- Data from scripts and processes
- Network streams, including monitoring of TCP, UDP, and HTTP traffic with the HTTP Event Collector
- Windows data, including Windows Event Log, Registry changes, and Performance Monitoring metrics

***Learn about how Splunk software uses input configurations to get data***

You can determine where data generation occurs after you have discovered your Splunk deployment topology. You can also do this while you are in the process of discovering your deployment topology, but it is easier to gather information on configurations after the deployment topology has been discovered.

Forwarders and indexers can get data input and other configurations in several ways:

- Locally, through an `inputs.conf` **configuration file**. This is the most common method for how Splunk instances get configuration information
- Through an **app** or **add-on** that has been installed on the instance
- From a **deployment server** that the forwarder or indexer has connected to

The deployment server is an advanced configuration topic outside the scope of this topic. To learn more about the deployment server and how it works, see About deployment server and forwarder management in *Updating Splunk Enterprise Instances*.

The `inputs.conf` file defines data inputs and controls aspects of data collection for the forwarder or indexer:

- When to collect data
- What type of data to collect
- How often to collect the data
- Where to index the data it has collected
- How to index the data it has collected

On forwarders, there is a file called `outputs.conf` that controls where the forwarder sends the data. Like `inputs.conf` it can be a standalone configuration, a configuration that is part of an app or add-on, or a configuration that has been retrieved from a deployment server.

Splunk software uses a scheme called configuration file precedence to build a master configuration file to handle multiple data collection and forwarding scenarios. See configuration file precedence in the *Admin Manual*.

### *Discover Splunk data collection configurations*

The following procedure represents high-level guidance for determining the inputs in your Splunk deployment.

1. After you locate indexers and forwarders in the deployment, confirm whether they have a local configuration for data inputs, get a configuration from an app or add-on, or retrieve configurations from a deployment server.
2. If the forwarder is configured to connect to a deployment server, check the deployment server to see its configurations. Any forwarder that connects to this server gets these configurations. The configurations can be standalone or contained within apps or add-ons.
3. Review `inputs.conf` configuration files to see what data is being collected. You can find these files in the following places:
    1. By themselves, in `$SPLUNK_HOME/etc/system/local`
    2. In an app or add-on, in `$SPLUNK_HOME/etc/apps/<name of app>/local`
    3. On a deployment server, in `$SPLUNK_HOME/etc/deployment_apps/<name of app>/local`
4. See the *Getting Data In Manual* for information about the types of data that each instance collects.
5. If you have a diagram of your Splunk deployment, indicate the locations of the data collecting instances in the diagram, and what data they are collecting.

## Next steps

After you have discovered where the data inputs are, you can do the following:

- Determine whether or not input configurations need to be added, changed, or removed, depending on business purpose or data collection performance improvements.
- Determine if you want to set up the Monitoring Console, if it has not already been set up
- Determine whether or not changes need to be made to index data according to Splunk best practices for getting data in.

# Review your apps and add-ons

If you inherited a Splunk Enterprise deployment for a large organization, you might have many apps and add-ons running on your system. This topic provides an overview of Splunk apps and add-ons, and helps you to identify the apps and add-ons installed on your Splunk Enterprise instance.

It is important to identify any Splunk Premium Solution apps running on your system. These apps provide comprehensive data analysis for specific use cases, such as IT operations and security, and can require additional resources and management.

## Splunk app and add-on overview

Splunk apps and add-ons are packaged sets of configuration files that you install on a Splunk Enterprise instance. Apps and add-ons are defined as follows:

**Apps**: Splunk apps provide user interfaces that let you work with your data. Apps often use one or more add-ons to ingest different types of data. See Apps and add-ons in the Splunk Enterprise *Admin Manual.*

**Add-ons**: Add-ons enable Splunk Enterprise, or a Splunk app, to ingest or map a particular type of data. See About Splunk add-ons in the *Splunk Add-ons* manual.

All Splunk apps and add-ons run on Splunk Enterprise. Splunk Enterprise includes the Splunk Search and Reporting app. This app provides the core search environment for Splunk Enterprise, and lets you create and manage Splunk **knowledge objects**, such as **saved searches**, **reports**, **alerts**, **dashboards**, **datasets**, and so on.

### *Deployment requirements and considerations*

Splunk apps and add-ons run on any supported Splunk Enterprise deployment topology, including single-instance, distributed, clustered, and cloud environments. To learn more about your existing Splunk Enterprise deployment topology, see Deployment topologies in this manual.

To familiarize yourself with any special requirements and considerations for your apps and add-ons, review the documentation for the specific app or add-on. To access documentation for all supported Splunk apps and add-ons, see Splunk Documentation.

Many Splunk apps are vetted for deployment on Splunk Cloud. If you encounter issues with app deployments in Splunk Cloud, or if you want to deploy additional apps to Splunk Cloud, contact Splunk support.

For more information, see:

- App deployment overview in the *Admin Manual*.
- Where to install Splunk Add-ons in the *Splunk Add-ons* manual.

## Survey your apps and add-ons

You can view all apps and add-ons installed on your system by using Splunk Web, which is the Splunk Enterprise UI, or by using the command line to navigate the file system on the search head.

### *View your apps and add-ons in Splunk Web*

The Manage apps page in Splunk Web gives you access to all apps and add-ons installed in your deployment. You can view information about the app, including app name, folder name, and version. You can also enable or disable the app, set app permissions using role-based access controls, and perform actions such as edit properties and view objects.

1. Open Splunk Web.
   All enabled apps appear in the App column at left.
2. Click **Apps > Manage Apps**.
   The Manage Apps page opens.
3. Review the list of apps and add-ons installed on your Splunk Enterprise instance.

For more information, see:

- View app and add-on objects in this topic.
- Edit app and add-on properties in the *Admin Manual.*

### *View your apps and add-ons using the file system on the Search Head*

1. Log in to a search head.
2. Navigate to the directory `$SPLUNK_HOME/etc/apps`.
   All of the apps and add-ons that are installed on your system are located in the `apps` directory.
3. Review your apps and add-ons.

### *App and add-on naming conventions*

Splunk app folder names use a variation or abbreviation of the app product name. The following table provides some examples.

| App name | App folder name |
|---|---|
| Splunk Enterprise Security | SplunkEnterpriseSecuritySuite |
| Splunk IT Service Intelligence | itsi |

| Add-on prefix | Add-on description | Example name |
|---|---|---|
| TA | Splunk technology add-on | Splunk_TA_stream |
| SA | Splunk supporting add-on | SA-ITOA |
| DA | Splunk domain add-on (ITSI module) | DA-ITSI-OS |

### *View app and add-on objects*

When you create an app or add-on, Splunk Enterprise creates a collection of objects that makes up the app or add-on. These objects can include views, commands, navigation items, event types, saved searches, reports, and so on.

In addition, each app object has role-based permissions associated with it that determine who can view or edit the object. By default, the Splunk Enterprise admin user has write permissions and can edit all objects across the system.

Use Splunk Web to view all objects that pertain to a specific app or add-on, as follows:

1. In Splunk Web, click **Settings > All configurations**.
2. In the **App context** menu, select the name of the app whose objects you want to view.
3. Select the **Show only objects created in this app context** check box.

For more information, see:

- Manage app and add-on objects in the *Admin Manual*.
- App architecture and object ownership in the *Admin Manual*.

### *Identify apps that use the KV store*

The **KV store** resides on every Splunk Enterprise version 6.2 or later instance by default and is often active on search heads. KV store can maintain state information about apps. In addition, some apps, like Enterprise Security, use the

KV store for lookups. KV store replicates its data across search heads using port 8191 by default. KV store processes are independent of a search head cluster's processes.

Discover KV store members using the Splunk command line interface. See About the CLI in the *Admin Manual*.

1. Log in to a search head.
2. Type `./splunk show kvstore-status`

Make note of the following:

- Whether disabled is 1 or 0.
- Which nodes are members of the KV store cluster.
- The port number that KV store is using.

Add the KV store members and port numbers to your deployment diagram. This command also returns information on which node is captain, but this information is not useful at this stage. Captaincy can change, so leave this detail off of your diagram.

Next, determine which apps, if any, use the KV store.

Apps that use the KV store have `collections.conf` defined in `$SPLUNK_HOME/etc/apps/<app name>/default`. In addition, `transforms.conf` has references to the collections with `external_type = kvstore`.

For a list of apps that have collections defined:

1. Log in to a search head.
2. At the command line, from the Splunk installation directory, type `./splunk btool collections list --debug`
3. In the results, look for items in `$SPLUNK_HOME/etc/apps`

For more information, see:

- Manage state with the key value store on the developer portal.
- Configure KV store lookups in the *Knowledge Manager Manual*.
- About the app key value store in the *Admin Manual*.
- KV store troubleshooting tools in the *Admin Manual*.

### *Identify deployment apps*

Distributed Splunk Enterprise deployments use the **deployment server** to distribute app and configuration file updates to groups of Splunk Enterprise components, such as forwarders, non-clustered indexers, and search heads. These apps and configuration files are called **deployment apps**. Deployment apps reside on a Splunk Enterprise instance that has been assigned the deployment server role, and are located in the directory `$SPLUNK_HOME/etc/deployment-apps`.

View your deployment apps in Splunk Web:

1. Identify which Splunk Enterprise instance is assigned the deployment server role. For help discovering the correct Splunk Enterprise instance, see Discover management components in this manual.
2. Log in to the deployment server.
3. Click **Settings > Forwarder management**.
4. On the Forwarder Management page, note the following:
    - **Apps**. Apps are the deployment apps currently being distributed by the deployment server.
    - **Clients**. Clients are the remote Splunk Enterprise instances to which the deployment server distributes the deployment apps.
    - **Server Classes**. Server classes are groups of deployment clients. The server class determines the specific set of clients that receive the app update.
5. Record server classes on your deployment diagram.

View your deployment apps using the file system on the deployment server:

1. Log in to the machine hosting the deployment server.
2. Go to `$SPLUNK_HOME/etc/deployment-apps`
3. Make note of the apps currently being distributed by the deployment server.

For more information, see:

- About deployment server and forwarder managment in *Updating Splunk Enterprise Instances*.
- Deploy Apps to client in *Updating Splunk Enterprise Instances*.

For information on deploying apps to **search head clusters**, **indexers**, and **indexer clusters**, see:

- Use the deployer to distribute apps and configuration updates in *Distributed Search*.
- Update common peer configurations and apps in *Managing Indexers and Clusters of Indexers*.

### *Download apps from Splunkbase*

Splunk offers a large number of apps and add-ons, free and for purchase, that can help you extend your data ingestion, search, and analysis capabilities. Splunk apps and add-ons are available for download at Splunkbase.

## Splunk Premium Solutions overview

Splunk Premium Solutions are apps developed by Splunk that provide comprehensive data search and analysis capabilities for specific use cases, such as IT operations analytics, and security threat detection and analysis.

Splunk offers the following Premium Solutions:

- Splunk Enterprise Security (ES)
- Splunk IT Service Intelligence (ITSI)
- Splunk User Behavior Analytics (UBA)

### *ES and ITSI requirements and considerations*

Splunk ES and ITSI production deployments can be resource intensive. Depending on several factors, such as the number of concurrent searches, the daily index volume, and the unused capacity of your environment, additional hardware might be required above the baseline Splunk Enterprise hardware. For the latest Splunk Enterprise hardware requirements, see Reference hardware in the Splunk Enterprise *Capacity planning manual*.

Familiarize yourself with the factors that affect ES and ITSI performance, including the respective number of correlation or KPI searches running and the number of concurrent users on the system. This will help you to evaluate the performance of your system and determine how and when to scale your deployment.

It is important to familiarize yourself with search head and indexer considerations that might impact the configuration of your deployment. For example, Splunk ES requires a dedicated search head, while ITSI does not.

For information on ES performance and capacity planning, as well as search head and indexer considerations, see Deployment planning in the Splunk Enterprise Security *Installation and Upgrade Manual*

For information on ITSI performance and capacity planning, as well search head and indexer considerations, see Deployment planning in the Splunk ITSI *Installation and Configuration Manual*.

## Splunk Enterprise Security overview

Splunk Enterprise Security (ES) detects patterns in your data and evaluates events for security-relevant incidents using correlation searches. When a **correlation search** detects a suspicious pattern, the correlation search can create a **notable event**. The app provides specialized dashboards and visualizations that you can use to you identify, triage, and analyze security incidents.

See the Splunk Enterprise Security documentation.

### *View your ES correlation searches*

View the correlation searches available in Splunk Enterprise Security and those that are enabled to better understand the use cases that Splunk Enterprise Security is being used to detect. To get a list of the correlation searches enabled in Splunk Enterprise Security, you can use a REST search to view the information in a table. See List correlation searches in Splunk Enterprise Security in *Administer Splunk Enterprise Security*.

### *The Content Profile dashboard*

Enterprise Security uses data that has been mapped to CIM-specific or ES-specific data models and accelerated to produce faster search results across a broad set of technologies. Review the data models in use in your environment and get an overview of the knowledge objects that correspond to the data models on the Content Profile dashboard. See Content Profile in *Use Splunk Enterprise Security*.

### *The Data Model Audit dashboard*

In addition, you can review the status of data models on the Data Model Audit dashboard and the retention and acceleration settings for data models. Data models that are not fully accelerated can result in missing or out-of-date information on dashboards or notable events in Splunk Enterprise Security. See Data Model Audit in *Use Splunk Enterprise Security* and Configure data models for Splunk Enterprise Security in the Splunk Enterprise Security *Installation and Upgrade Manual*.

### *Learn more about Splunk Enterprise Security*

To learn more about important Splunk Enterprise Security concepts and features, see:

- Incident review in *Administer Splunk Enterprise Security*.
- Correlation search overview in *Administer Splunk Enterprise Security*.
- Add asset and identity information to Splunk Enterprise Security in *Administer Splunk Enterprise Security*.
- Add threat intelligence to Splunk Enterprise Security in *Administer Splunk Enterprise Security*.
- Risk Analysis in *Use Splunk Enterprise Security*.
- Accelerate your investigations with security intelligence in *Use Splunk Enterprise Security*.
- Monitor security domain activity in *Use Splunk Enterprise Security*.

## Splunk IT Service Intelligence overview

Splunk IT Service Intelligence (ITSI) monitors the health of IT services using key performance indicators (KPIs) that track the severity-level of IT performance metrics. When KPI values meet threshold conditions, ITSI generates a notable event. The app provides features for aggregating and analyzing notable events, as well as dashboards and visualizations that let you continuously monitor IT services and perform root cause investigations.

See the Splunk IT Service Intelligence documentation.

### *View your ITSI services and service KPIs*

Review your services and the KPIs that your services contain. This will help you understand the IT operations and business processes that your services are monitoring, and it will help you identify the performance metrics being used to evaluate service health. It will also help you understand KPI search properties, including source search types (data model, ad hoc, or base search), calculations (search frequency and calculated stat), and severity-level thresholds that determine the KPI health status.

To view your services and service KPIs:

1. In the ITSI main menu, click **Configure > Services**.
2. Review the list of services.
3. Click on any service. For example, Database Service. The service configuration workflow appears.
4. Review the list of KPIs contained by the service. Each KPI represents an IT performance metric, such as CPU Utilization%, Memory Free %, Response Time, and so on.
5. Click on any KPI in the list.
6. Open the Search and Calculate panel.
7. For Source, note the **Threshold field**. This is the field in your data for which the KPI search returns a value. For example, cpu_load_percent. Click **Edit** to examine the source search details. Note that base searches, such as those provided by ITSI modules, tend to provide best search performance.
8. For Entities, note the entity alias filter settings. This determines the entities against which a KPI search runs.
9. For Calculations, note the stat that the KPI calculates, for example Average. Also note the KPI frequency and time range. KPIs can run every 1, 5, or 15 minutes.
10. Open the Threshold panel.
11. In the threshold preview graph, note the severity-level thresholds set for the KPI. When KPI values meet threshold conditions, the KPI health status changes, for example, from high to critical.

For more information, see:

- Create Services in the ITSI *Installation and Configuration Manual*.
- Configure ITSI Services in the ITSI *Installation and Configuration Manual*.

### *Review associated entities*

Identify the entities associated with your services. Entities are IT components that act as the primary data sources for ITSI services. KPI searches run against entities based on filtering conditions that you define. In more complex ITSI deployments a single entity can be associated with multiple services and have multiple different KPIs running against it.

To view entities associated with a service:

1. In the ITSI main menu, click **Configure > Entities**.
2. Review the list of entities. In the services column, note the services associated with each entity.
3. For any entity in the list, click **View Health**.
4. Review the entity details page, which shows all of the services the entity is associated with, and all of the KPIs running against the entity.

For more information, see Define Entities in the ITSI *Installation and Configuration Manual*.

### *View all ITSI KPIs*

Use Splunk Web to view all KPI searches running on the search head. This will give you an idea of the number of concurrent searches contributing to the search load. You can view additional information, including the KPI search string, search frequency, time range, and run times for recent KPI search jobs.

1. In Splunk Web, click **Setttings > Search, reports, and alerts.**
2. Select the **Show only objects created in this app context** checkbox.

   All apps created in the ITSI app context appear in the list. KPI search names use the following syntax:

   ```
   Indicator - <KPI_id> - ITSI Search
   ```

   For example

   ```
   Indicator - 3bee62acf7f4de2a095e475f - ITSI Search
   ```
3. For any KPI search, click **View Recent**. Note the KPI run time.
4. Click on the name of the KPI search. Note the KPI search string, time range, and schedule.

Be aware that average KPI run time, KPI frequency, and the number of entities referenced per KPI, along with the total number of concurrent searches running on the system can markedly impact performance. For more information, see Performance considerations in the ITSI *Installation and Configuration Manual*.

### *Learn more about Splunk ITSI*

To learn more more about Splunk ITSI, see ITSI concepts and features in the ITSI *Installation and Configuration Manual*.

## About Splunk User Behavior Analytics

Splunk User Behavior Analytics (UBA) helps you find known, unknown, and hidden threats in your environment. You can use Splunk UBA to visualize and investigate internal and external threats and anomalies. Splunk UBA integrates with Splunk Enterprise Security to take advantage of Splunk events and to investigate UBA threats alongside other notable events in your organization.

See the Splunk User Behavior Analytics documentation.

# Users, roles, and authentication

Once you have familiarized yourself with your Splunk configuration and data, review your users, their permissions, and their authorization methods.

Splunk Enterprise supports several user authentication systems:

- Splunk internal authentication with role-based user access
- LDAP
- A scripted authentication API for use with an external authentication system, such as PAM or RADIUS
- Multifactor authentication
- Single sign-on

## Internal authentication and role-based user access

Role-based access control lets you manage users and restrict or share Splunk Enterprise data. Splunk Enterprise masks data to users in the similar manners that a relational database manages role-based access control.

### *Discover or modify existing configurations*

Familiarize yourself with your existing users and their assigned roles. Roles determine the user's data access level and the actions they can perform.

In Splunk Web click **Settings** > **Access Controls** to see all of your Splunk users. On the Access Controls page you can click on roles and users to examine or edit permissions. You can use this page to create a list of the data available to each user or group of users. See Use access control to secure Splunk data in *Securing Splunk Enterprise*.

To find a specific user you can use the CLI to search for a user and role. See Find existing users and roles in *Securing Splunk Enterprise*.

## LDAP authentication

When administrators configure Splunk to work with LDAP, they create something called "LDAP strategies". LDAP strategies are collections of configuration data that Splunk uses to work with your LDAP configuration. Splunk can be directed to query these "strategies" in a particular order when searching for LDAP users. See Set up user authentication with LDAP in *Securing Splunk Enterprise*.

### *Discover or modify existing LDAP configurations*

Familiarize yourself with the existing LDAP groups and permissions mappings by looking at all of your strategies. To view or edit existing LDAP strategies, follow these steps:

1. Under **Users and authentication** click **Access controls**.

2. Click **LDAP**.

3. From this page, you can select strategies and view their information and track those LDAP mappings to Splunk roles.

See Configure LDAP with Splunk Web in *Securing Splunk Enterprise*.

## Multifactor authentication

Splunk Enterprise currently supports multifactor authentication with Duo Security. See About two-factor authentication with Duo Security in *Securing Splunk Enterprise*.

### *Find or modify existing configurations*

Find out if your system uses Duo Factor Authentication via Splunk Web.

1. Under **Settings** click **Users and Authentication**

2. For **Authentication Method** select **Duo Security**.

3. On this page you can see if your system has mutifactor authentication configured. See Configure Splunk Enterprise to use Duo Security two-factor authentication in *Securing Splunk Enterprise*.

## SSO with SAML

Splunk software can leverage SAML authentication for single sign-on (SSO), using information provided by an external identity provider (IdP). See Authentication using single sign-on with SAML in *Securing Splunk Enterprise*.

### *Find or modify existing configurations*

Find out if your users are configured for SAML SSO.

1. In **Settings** select **Access Controls**.

2. Under **Authentication method** select SAML.

3. A new SAML configuration appears, you can close this page to view the existing configuration.

In this page you can see if your system has SSO authentication configured for groups of users. From there you can drill down to your IdP information, the mapped groups, and the users assigned to that group.

## ProxySSO authentication

ProxySSO lets you configure Single-Sign On (SSO) for Splunk instances through a reverse proxy server. A user logged in using ProxySSO can seamlessly access Splunk Web.

### *Find existing configurations*

You can view any existing HTTP request headers that the proxy server sends to Splunk Web:

Set `enableWebDebug=true` in `web.conf` under `settings` stanza:

```
http://<ProxyServerIP>:<ProxyServerPort>/debug/sso
```

ProxySSO login events are logged in `var/log/splunkd.log`.

## Review your system security

Splunk software ships with a set of default certificates. The default certificates are generated and configured at startup and can be found in `$SPLUNK_HOME/etc/auth/`. Splunk recommends that administrators replace these default certificates with self- or third-party-signed certificates.

The following table describes the most common scenarios and the default SSL settings.

| Type of exchange | Client function | Server function | Encryption | Certificate Authentication | Common Name checking | Type of exchan |
|---|---|---|---|---|---|---|
| Browser to Splunk Web | Browser | Splunk Web | NOT enabled by default | dictated by client (browser) | dictated by client (browser) | search ter results |
| Inter-Splunk communication | Splunk Web | `splunkd` | enabled by default | NOT enabled by default | NOT enabled by default | search ter results |
| Forwarding | `splunkd` as a forwarder | `splunkd` as an indexer | NOT enabled by default | NOT enabled by default | NOT enabled by default | data to be indexed |
| Deployment server to indexers | `splunkd` as a forwarder | `splunkd` as an indexer | NOT enabled by default | NOT enabled by default | NOT enabled by default | Not recommen Use Pass4Sym instead. |
| Inter-Splunk communication | `splunkd` as a deployment client | `splunkd` as deployment server | enabled by default | NOT enabled by default | NOT enabled by default | configurati data |
| Inter-Splunk communication | `splunkd` as a search head | `splunkd` as search peer | Enabled by default | NOT enabled by default | NOT enabled by default | search dat |

### *Verify your SSL configurations*

**Splunk Web**

Use the following command to verify your SSL connections in Splunk Web:

```
index=_internal source=*metrics.log* group=tcpin_connections | dedup
hostname | table _time hostname version sourceIp destPort ssl
```

**Indexer and forwarder**

On the indexer, look for the following or similar messages at the start-up sequence to verify a successful connection:

```
02-06-2011 19:19:01.552 INFO TcpInputProc - using queueSize 1000
02-06-2011 19:19:01.552 INFO TcpInputProc - SSL
cipherSuite=ALL:!aNULL:!eNULL:!LOW:!EXP:RC4+RSA:+HIGH:+MEDIUM
02-06-2011 19:19:01.552 INFO TcpInputProc - supporting SSL v2/v3
02-06-2011 19:19:01.555 INFO TcpInputProc - port 9997 is reserved for
splunk 2 splunk (SSL)
02-06-2011 19:19:01.555 INFO TcpInputProc - Port 9997 is compressed
02-06-2011 19:19:01.556 INFO TcpInputProc - Registering metrics callback
for: tcpin_connections
```

On the forwarder, look for the following or similar messages at the start-up sequence to verify a successful connection:

```
02-06-2011 19:06:10.844 INFO TcpOutputProc - Retrieving configuration
from properties
02-06-2011 19:06:10.850 INFO TcpOutputProc - Using SSL for server
10.1.12.112:9997, clientCert=/opt/splunk/etc/aut/server.pem
02-06-2011 19:06:10.854 INFO TcpOutputProc - ALL Connections will use
SSL with sslCipher=
02-06-2011 19:06:10.859 INFO TcpOutputProc - initializing single
connection with retry strategy for 10.1.12.112:9997
```

Following is how a successful connection might appear in `splunkd.log` on the indexer:

```
02-06-2011 19:19:09.848 INFO TcpInputProc - Connection in cooked mode
from 10.1.12.111
02-06-2011 19:19:09.854 INFO TcpInputProc - Valid signature found
02-06-2011 19:19:09.854 INFO TcpInputProc - Connection accepted from
10.1.12.111
```

Following is how a successful connection might appear in `splunkd.log` on the forwarder:

```
02-06-2011 19:19:09.927 INFO TcpOutputProc - attempting to connect to
10.1.12.112:9997...
02-06-2011 19:19:09.936 INFO TcpOutputProc - Connected to
10.1.12.112:9997
```

## About securing distributed environments

Communication between search heads and peers uses public-key encryption.

At startup, Splunk software generates a private key and a public key on your Splunk installation. When you configure distributed search on the search head, the public keys are distributed by search heads to peers and those keys are used to secure communication. This default configuration provides built-in encryption as well as data compression that improves performance. See Distribute the key files in the *Distributed Search Manual*.

Public-key encryption for securing distributed configurations. However, it is possible to configure SSL for a search head cluster by configuring each member of the search head cluster. You can determine if your deployment has each member of the search head cluster configured for SSL by checking the attribute `requireClientCert` in `server.conf`. See Secure your deployment server and clients using certificate authentication in *Securing Splunk Enterprise*.

## Encryption with the `splunk.secret` key

The `splunk.secret` file contains a key that collects and encrypts some of your authentication information in configuration files:

- `web.conf`: SSL passwords on every instance
- `authentication.conf`: LDAP passwords, if you have any
- `inputs.conf`: SSL passwords, if you use `splunktcp-ssl`
- `outputs.conf`: SSL passwords, if you use `splunktcp-ssl`
- `server.conf`: pass4symmkey, if you have one

At initial startup, Splunk Enterprise creates this file at `$SPLUNK_HOME/etc/auth/`. Any passwords you create in the above list are stored in this file. If you manually add any unencrypted passwords, Splunk software will overwrite those passwords upon startup.

## More information

- About securing Splunk Web

- About securing data from forwarders

- About securing Splunk to Splunk communication

- Secure Splunk Web with your own certificate
- Configure Splunk forwarding to use your own certificates
- About securing inter-splunk communication

# Learn about licensing

## How Splunk Enterprise licensing works

Splunk Enterprise takes in data from sources that you designate and processes it so that you can analyze it. This process is called indexing. For information about the indexing process, see How Splunk software handles your data in *Getting Data In*.

Splunk Enterprise licenses specify how much data you can index per calendar day (from midnight to midnight by the clock on the **license master**).

Any Splunk Enterprise instance that performs indexing must be licensed to do so. You can either run a standalone indexer with a license installed locally, or you can configure one of your Splunk Enterprise instances as a **license master** and set up a **license pool** from which other indexers, configured as **license slaves**, can draw.

If you exceed your licensed daily volume on any one calendar day, you get a violation *warning*. If you have 5 or more warnings on an Enterprise license in a rolling 30-day period, you are in *violation* of your license. Unless you are using a Splunk Enterprise 6.5.0 or later no-enforcement license, search is disabled for the offending **pool** or pools. Other pools remain searchable, as long as the total license usage from all pools is less than the total license quota for the license master.

In addition to indexing volume, access to some Splunk Enterprise features requires an Enterprise license.

There are a few types of licenses, such as:

- The Enterprise license enables all Enterprise features, such as authentication and distributed search. As of Splunk Enterprise 6.5.0, new Enterprise licenses are no-enforcement licenses.

- The Free license allows for a limited indexing volume and disables some features, including authentication.
- The Forwarder license allows you to forward, but not index, data, and enables authentication.
- The Beta license typically enables Enterprise features, but is restricted to Splunk Beta releases.
- A license for a premium app is used in conjunction with an Enterprise or Cloud license to access the functionality of an app.

For more information about different types of licenses, read Types of Splunk licenses in the *Admin Manual*.

## Understand your licenses

Survey what licenses you have:

1. Log into Splunk Web on your license master.
2. Click **Settings > Licensing**.
3. Make note of Enterprise and app licenses and their expiration dates.

Check your license usage:

1. Log into Splunk Web on your license master.
2. Click **Settings > Licensing**.
3. Click **License usage report**.

See About the license usage report view in the *Admin Manual*. This view is also accessible from the **Indexing** tab of the monitoring console.

## Monitor your license usage

To prevent license violations, set up alerts for expiring licenses and licenses nearing quota. You can use the two platform alerts included with the monitoring console.

For more information, see:

- Platform alerts in *Monitoring Splunk Enterprise*.
- About license violations in the *Admin Manual*.

### Update Support contact

Splunk licenses are tied to your organization's customer account at Splunk. A customer account typically has one or several employees that are authorized to received Splunk Support after some training, in addition to a Splunk portal administrator who manages the list of authorized contacts. Make sure you understand how your organization contacts Support, before you need to.

# Monitor system health

If you have the monitoring console configured, you can use platform alerts and the health check to monitor your system health.

If you do not have the monitoring console configured, this topic directs you to a few tools to get started. But if you have read and understand the content in the previous several topics, it might be time to think about setting up the monitoring console.

## With the monitoring console

#### *Run a health check*

The monitoring console comes with preconfigured health checks in addition to its preconfigured platform alerts. You can modify existing health checks or create new ones.

See Access and customize health check in *Monitoring Splunk Enterprise*.

#### *Understand platform alerts*

Platform alerts are saved searches included in the Monitoring Console. Platform alerts notify Splunk Enterprise administrators of conditions that might compromise their Splunk environment. Notifications appear in the Monitoring Console user interface and can optionally start an alert action such as an email.

See which platform alerts are enabled:

1. In the monitoring console, click **Overview**.
2. Scroll down the dashboard until you see the Triggered Alerts panel.
3. Click **Enable or Disable** to get to the Platform Alerts Setup page.
4. Look for an enabled alert.

5. Click **Advanced edit** to see what alert actions exist for that alert. Add your email address if you want to receive email alerts. If you do not set up an alert action like **Send an email**, view any triggered alerts in the monitoring console Overview dashboard.

See Platform alerts in *Monitoring Splunk Enterprise* for instructions for adding alert actions and for a list of available platform alerts.

### *Rebuild the forwarder asset table*

If a forwarder is decommissioned, it remains on the forwarder dashboards until you rebuild the forwarder asset table. This step might not be immediately necessary, but if you find that your forwarder dashboards contain null results from several forwarders, you can rebuild the asset table.

1. In the monitoring console, click **Settings > Forwarder Monitoring Setup**.
2. Click **Rebuild forwarder assets**.
3. Select a time range or leave the default of 24 hours.
4. Click **Start Rebuild**.

## Without the monitoring console

### *Ensure that internal logs are searchable*

Make sure that your deployment is following the best practice recommendation of forwarding internal logs, in both `$SPLUNK_HOME/var/log/splunk` and `$SPLUNK_HOME/var/log/introspection`, to indexers from all other instance types. See Best practice: Forward search head data in the *Distributed Search Manual*. These other instance types include:

- search heads
- license masters
- cluster masters
- deployment servers

See What Splunk software logs about itself in *Troubleshooting Splunk Enterprise* for an overview of the Splunk Enterprise internal log files.

### *Survey existing monitoring apps*

Survey your deployment for apps monitoring system health, either from Splunkbase or a custom app that a previous administrator developed.

- The Fire Brigade app gives you insight into the health of your indexers.
- The popular Splunk on Splunk app (SoS) reached its end of life with Splunk Enterprise 6.3.0 and most of its functionality was incorporated into the monitoring console. If your monitoring strategy relies on SoS, consider upgrading to the monitoring console.

### *Use default monitoring tools*

Even without the monitoring console, there are a few resources that are included in Splunk Enterprise for you to check your system health. You can view some status information about indexer clustering, search head clustering, KV store, and errors that Splunk software logs internally.

For information on indexer clustering dashboards, see View the master dashboard and the two following topics in *Managing Indexers and Clusters of Indexers*.

You can run status checks on portions of your deployment using the Splunk command line, like search head clustering and KV store.

You can check components of a search head cluster from the command line. See Use the CLI to view information about a search head cluster in the *Distributed Search* manual.

Check KV store status:

1. Log into a search head.
2. In a terminal window, navigate to the bin directory in the Splunk installation directory.
3. Type `./splunk show kvstore-status`

See About the CLI in the *Admin Manual* for information about using the Splunk CLI.

Generate a report of general errors:

1. Log into Splunk Web on a cluster master or search head.
2. Click **Apps > Search & Reporting**.
3. Click **Reports > Splunk errors last 24 hours**.

### *Look for custom monitoring tools*

In addition to a custom monitoring app, your previous admin might have created some custom reports or alerts for system health. Look for custom reports or alerts:

1. In Splunk Web on a cluster master or search head, go to **Settings > Searches, reports, and alerts**.
2. Review any alert actions and make sure that they meet your requirements.
3. (Optional) Add your email address or a custom script.

### *Plan a monitoring strategy*

Any production Splunk Enterprise deployment requires robust, proactive monitoring to minimize down time and other problems.

Your Splunk Enterprise monitoring strategy needs to address the following points, the monitoring of which is included in the monitoring console:

- CPU load, memory utilization, and disk usage
- On a *nix system, OS level settings such as THP and ulimits
- Indexing rate
- Skipped searches
- Bad data onboarding practices

Consider setting up the monitoring console. More than likely, this will consist of provisioning a new machine for this use. See Multi-instance deployment Monitoring Console setup steps in *Monitoring Splunk Enterprise*.

## Investigate knowledge object problems

**Knowledge objects** are user-defined entities that enrich your existing data. They include the following objects:

- **reports**
- **alerts**
- **dashboards**
- **datasets**
- **field extractions**
- **calculated fields**
- **event types**

- **lookups**
- **tags**
- **aliases**

You manage most knowledge objects through their listing pages in the Search and Reporting view, or through the pages listed under **Knowledge** in the **Settings** menu.

Organizations with large Splunk Enterprise deployments often have **knowledge managers**, people whose roles consist of creating, organizing, and maintaining knowledge objects for other Splunk Enterprise users. See the *Knowledge Manager Manual*.

## Survey your knowledge object landscape

Review the knowledge object collections in your Splunk Enterprise deployment. You can use the knowledge object pages in **Settings** to review each knowledge object category across all of your installed apps. For example, if you want to look at your saved searches, select **Settings > Searches, Reports, and Alerts**.

As you review your knowledge objects, note their names, app affiliations, owners, and permission status. Identify knowledge objects that have naming or permissions conflicts, are redundant, or are orphaned.

## Knowledge object naming conflicts

As you review a category of knowledge objects, look for two types of nomenclature conflicts:

- Objects that share the same name but which have different definitions.
- Objects that share the same definition, but which have different names.

### *Same name, different definitions*

All objects within a knowledge object category must have unique names. For example, there can be no duplicate names among the saved searches in the Searches, reports, and alerts listing page in **Settings**. Most of these knowledge objects are applied to your search results at search-time. If you have more than two objects of the same category with the same name, only one of those objects is applied.

Duplicate naming can happen when objects have their permissions changed. For example, you can have lookups in two separate apps that have the same name.

They do not conflict with each other when they are shared at the app level. However, if one of those lookups has their permissions changed so that they are shared globally, it is possible for one of those lookups to be applied instead of the other.

See Give knowledge objects the same names in the *Knowledge Manager Manual*.

Avoid this problem by establishing naming conventions. See Develop naming conventions for knowledge objects in the *Knowledge Manager Manual.*

***Same definition, different names***

If you have multiple knowledge objects in a category that have the same or similar definitions, but different names, you have a normalization problem. This can especially be a problem with extracted fields. When you index data from multiple source types, you can have several fields with different names but which represent the same kind of data. This leads to a misunderstanding of your indexed data. You might inadvertently build searches that account for a portion of the information that you want to capture.

If your Splunk Enterprise deployment has data normalization problems, install the Splunk Common Information Model Add-on. The CIM Add-on can help you to normalize the data from multiple source types so that you can develop reports, correlation searches, and dashboards that present unified views of your data domains.

See the *Splunk Common Information Add-on Manual*.

## Understand your object permissions

As you manage the knowledge objects that you have inherited, ensure that you understand how **roles**, **capabilities**, and **permissions** are set up for your Splunk deployment.

When a user creates a knowledge object, its permissions are private to that user by default. Depending on how your Splunk Enterprise deployment is set up, that user may need to rely on someone with an admin or power user role to share that object with other users and roles.

### *Permissions and knowledge object interdependencies*

If all of your objects have the same permissions, it is easy to resolve dependency issues between knowledge objects. For example, you can have a private scheduled report that uses the `outputlookup` command to update widely-used lookups with global permissions. Over time your users may find that the lookup is behaving unpredictably, but because private knowledge objects are invisible to most users the cause of the problem can be hard to troubleshoot and resolve.

For more examples, see Object interdependency considerations.

### *Other uses of permissions*

There are more aspects to permissions than expanding or restricting the visibility of knowledge objects. You can use the permissions features for the following tasks:

- Use role-based capabilities to restrict or expand the ability to create and edit knowledge objects.
- Enable roles other than Admin and Power User to set permissions and share objects.
- Set permissions for knowledge object categories. For example, you can restrict the ability of certain roles to use all event types or all lookups.

To learn more about roles and capabilities see About configuring role-based user access in the *Securing Splunk Enterprise* manual.

To learn more about knowledge object permissions, see Manage knowledge object permissions in the *Knowledge Manager Manual*.

## Object interdependency considerations

There can be significant interdependencies between groups of objects. An object change or deletion can affect other objects that are dependent on that object. For example, you can have a lookup with a definition that references a custom field extraction. If you change how that field is extracted, it can affect the accuracy of the lookup. If that lookup is used to add fields to a data model dataset, the change will cascade down through all of its child data model datasets.

In many cases the only way to uncover knowledge object interdependencies is by studying your object definitions, or by analyzing the downstream object breakages that occur when upstream objects are changed, disabled, or deleted. See Disable or delete knowledge objects in the *Knowledge Manager Manual*.

### *The sequence of search time operations*

If you have interdependent knowledge objects, it is important to understand the sequence of search-time operations. At search time, the Splunk software applies knowledge objects to the results of a search in a specific order. This means that you cannot set up knowledge object interdependencies that depend on objects that have not yet been defined. If you find that some of your interdependent knowledge objects do not work, this is a possible cause.

For example, the Splunk software applies custom field extractions to search results before it processes lookups. This means that a lookup can have a definition that refers to a field that is extracted at search time. However, a custom field extraction cannot use a lookup-derived field in its definition, because that lookup field does not yet exist. It is derived only after the custom field extraction is processed.

See The sequence of search-time operations in the *Knowledge Manager Manual*.

### *Lookup object interdependencies*

**Lookups** can involve knowledge object interdependency by design.

Following are the three knowledge object categories related to lookups:

- Lookup definitions
- Lookup table files
- Automatic lookups

Any object from these categories can be assigned its own permissions and sharing status.

You can utilize those knowledge object categories to create the following lookup types:

- CSV lookups
- External lookups
- KV store lookups
- Geospatial lookups

All lookup types require a lookup definition. Two of the lookup types, CSV and geospatial, require a lookup table file. You can optionally create an automatic lookup for all lookup types.

Use caution when deleting or modifying lookup objects. A lookup table file can be associated with multiple lookup definitions. A lookup definition can also be associated with multiple automatic lookups.

You can run into permissions issues with lookups. If a lookup table file has permissions that are more restrictive than the definitions that it is associated with, the lookup does not work. The same is true for lookup definitions and automatic lookups.

- The permissions of a lookup table file should be wider than or equal to the permissions of the lookup definitions it is associated with.
- The permissions of a lookup definition should be wider than or equal to the permissions of the automatic lookups it is associated with.

See Introduction to lookup configuration in the *Knowledge Manager Manual*.

### *Data model dataset hierarchies*

Data models can be hierarchically-organized collections of data model datasets with parent/child relationships. A change to a parent dataset will cascade down through all of the child datasets that are descended from it. You can view these relationships with the **Data Model Editor**.

See Design data models in the *Knowledge Manager Manual*.

### *Dataset extension*

All dataset types, lookup, data model, and table, can be **extended** as **table datasets**. When extended, the original dataset has a parent relationship to the table dataset that is extended from it. A change to the original dataset affects any datasets that are extended from it. You can see what datasets a dataset is extended from by expanding its row on the Datasets listing page.

See Dataset extension in the *Knowledge Manager Manual*.

## Find orphaned objects

When the Splunk account of a knowledge object owner is deactivated, the knowledge objects that they owned remain in the system. These objects are orphaned, and they can cause problems. Orphaned objects can adversely affect objects that they are interdependent with.

Orphaned scheduled reports are especially problematic. The search scheduler cannot run a scheduled report on behalf of a nonexistent user. This affects dashboard panels and embedded reports that use the scheduled report. If the results of the report are sent by email to stakeholders, those emails cease.

Splunk Enterprise provides several methods of detecting orphaned knowledge objects, especially orphaned scheduled searches. When you find orphaned knowledge objects, you can use the Reassign Knowledge Objects page to reassign one or more of those knowledge objects to a new owner.

See Manage orphaned knowledge objects in the *Knowledge Manager Manual*.

## Scheduled searches and search concurrency

If your Splunk Enterprise deployment depends on a large number of scheduled reports and alerts, check whether it is encountering search concurrency issues.

All Splunk Enterprise deployments have limits to the number of scheduled searches that can run concurrently. Once this limit is reached, a background process called the **search scheduler** prioritizes the excess reports and runs them as other scheduled reports and alerts complete their runs.

The goal of the search scheduler is to have each scheduled report and alert run at some point within their periods, over the time ranges they were originally scheduled to run for. But you can encounter situations where certain reports regularly skip their scheduled runs.

### *Use the Monitoring Console to identify scheduler issues*

You can use the Monitoring Console to identify searches that are frequently skipped, or that are causing other searches to be frequently skipped. You can also use the monitoring console to see your system-wide concurrent search limits. See Scheduler activity in *Monitoring Splunk Enterprise*.

### *Reduce the number of skipped reports*

You can reduce the number of skipped scheduled reports by applying either the **Schedule Window** or **Schedule Priority** settings to scheduled reports. These settings are mutually exclusive. Apply a **Schedule Window** to low-importance report to enable other reports to run ahead of them. Use **Schedule Priority** to improve the run priority of high-value reports. See Prioritize concurrently scheduled reports in Splunk Web in the *Knowledge Manager Manual*.

## Report, data model, and dataset acceleration

You might have inherited a Splunk Enterprise deployment that uses acceleration to improve the performance of reports, data models, and table datasets. Your deployment might include apps that are delivered with accelerated data models and reports by default, such as Splunk Enterprise Security, Splunk IT Service Intelligence, and the Common Information Model Add-on .

If your deployment has other objects accelerated beyond the ones provided by your apps and add-ons, verify that they are functioning correctly and determine whether their summaries are needlessly using valuable disk space.

On the listing pages for reports, data models, and table datasets, acceleration is indicated by a yellow lightning bolt symbol.

For an overview of the summary-based acceleration options offered by Splunk, see Overview of summary-based search acceleration in the *Knowledge Manager Manual*.

### *Review report acceleration summaries*

You can access report acceleration summary statistics by selecting **Settings > Report acceleration summaries**.

| Action | Details |
|---|---|
| Identify unnecessary summaries. | On the Report Acceleration Summaries page in Settings, look for summaries with a high **Summarization Load** and a low **Access Count**.<br>Consider removing these summaries. The stats indicate that they are using a lot of system resources but are not being accessed often.<br>You can click through to the Summary Details of a particular summary to see its **Size on Disk**. Infrequently-used summaries that take up a lot of space are also good removal candidates. |
| Identify dysfunctional summaries. | On the Report Acceleration Summaries page in Settings, if the **Summary Status** is **Suspended** or **Not enough data to summarize**, the summary might have problems that need to be resolved. The Splunk software might not create summaries when |

| Action | Details |
|---|---|
| | it projects that they will be too large. |
| Resolve summary issues. | 1. Select **Settings > Report acceleration summaries**.<br>2. Find the summary that should be removed and open its detail page by clicking its **Summary ID**.<br>3. (Optional) Click **Verify** if you suspect the summary contains inconsistent data.<br>4. (Optional) Click **Update** if the summary has not been updated in some time and you want to make it current.<br>5. (Optional) Click **Rebuild** to rebuild summaries that fail verification or which seem to have data loss issues. Rebuilds of significantly large summaries can be time-consuming. |
| Delete summaries that are unnecessary. | 1. Select **Settings > Report acceleration summaries**.<br>2. Find the summary that should be removed and open its detail page by clicking its **Summary ID**.<br>3. Click **Delete** to remove the summary. |

See Manage report acceleration in the *Knowledge Manager Manual*.

### *Investigate data model and dataset summaries*

You can manage acceleration for data models and table datasets through the Data Models page in Settings. Expand the rows of the data models and table datasets to see their stats.

Look for summaries with low **Access Count** numbers and high **Size on Disk** numbers. Consider removing these summaries, or reducing their summary windows so that they do not take up as much disk space.

If you have summaries with build processes that are not completing, refer to Accelerate data models in the *Knowledge Manager Manual*. It discusses advanced configurations that can help you resolve these issues.

### *Review size-based summary retention rules*

A deployment that includes size-based retention rules for report, data model, and table dataset summaries can use an unlimited amount of disk space. Your deployment might have configured size-based retention rules to prevent that.

Review these configurations, identify the summaries that are affected by them, and evaluate whether the configurations need to be updated or removed.

For information about report acceleration summary retention configurations, see Manage report acceleration in the *Knowledge Manager Manual*.

For information about data model and table dataset summary retention configurations see Accelerate data models in the *Knowledge Manager Manual*.

### *Check parallel summarization for data models and table datasets*

Parallel summarization is a background process that increases the speed with which the Splunk software builds acceleration summaries for data models and table datasets. It does this by running concurrent searches to build the summaries. It is enabled by default for all Splunk Enterprise deployments.

If you are encountering persistent search concurrency or search performance issues, check whether your predecessor has raised the parallel summarization setting above its default. It might be running more concurrent searches than your system can support.

See Accelerate data models in the *Knowledge Manager Manual*.

### *Evaluate your summary indexes*

Summary indexing is a report acceleration method that you can use for reports that do not qualify for report acceleration. If your deployment uses summary indexing, it has indexes that are used specifically for summary indexing. Review these summary indexes and the searches that populate them with data, and evaluate whether they can be replaced by report acceleration summaries.

See Overview of summary-based search acceleration in the *Knowledge Manager Manual*.