# Table of Contents

## Introduction
- Monitoring Splunk Enterprise overview: 1

## About the Monitoring Console
- About the Monitoring Console: 2
- What can the Monitoring Console do?: 3
- How the Monitoring Console works: 3

## Configure the Monitoring Console
- Multi-instance Monitoring Console setup steps: 5
- Single-instance Monitoring Console setup steps: 5
- Which instance should host the console?: 6
- Monitoring Console setup prerequisites: 8
- Add Splunk Enterprise instances to the Monitoring Console: 10
- Configure Monitoring Console in standalone mode: 11
- Configure the Monitoring Console in distributed mode: 11
- Configure forwarder monitoring for the Monitoring Console: 12
- Enable and configure platform alerts: 13
- Access and customize health checks: 15

## Monitoring Console dashboard reference
- Indexing: Performance: 19
- Indexing: Indexes and Volumes: 20
- Indexing: Inputs: HTTP Event Collector: 21
- Indexing: Inputs: Data Quality: 21
- Indexing: License Usage: 22
- Indexing: SmartStore: 22
- Indexing: Indexer Clustering: Status: 23
- Indexing: Indexer Clustering: Service Activity: 23
- Search: Search Activity: 24
- Search: Search Usage Statistics: 26
- Search: KV Store: 27
- Search: Scheduler Activity: 29
- Search: Distributed Search: 31
- Search: Search Head Clustering: 32
- Resource Usage: 34
- Forwarders: 36

## Proactive Splunk component monitoring
- About proactive Splunk component monitoring: 38
- Configure the splunkd health report: 40
- Set up alerts for the splunkd health report: 45
- Set access controls for the splunkd health report: 49
- Investigate feature health status changes: 50
Introduction

Monitoring Splunk Enterprise overview

Splunk Enterprise provides two types of deployment monitoring:

- Monitoring Console (search-based monitoring).
- Proactive Splunk component monitoring (REST-based monitoring).

Monitoring Console

The Monitoring Console is a search-based monitoring tool that lets you view detailed information about the topology and performance of your Splunk Enterprise deployment. The Monitoring Console provides pre-built dashboards that give you visibility into many areas of your deployment, including search and indexing performance, resource usage, license usage, and more. You can use the Monitoring console to track the status of all types of deployment topologies, from single-instance (standalone) deployments to complex multi-site indexer clusters.

For more information, see About the Monitoring Console.

Proactive Splunk component monitoring

Proactive Splunk component monitoring is a REST-based monitoring tool that lets you view the health of Splunk Enterprise features from the output of a REST API endpoint. Individual features report their health status through a tree structure that provides a continuous, high-level view of the health of your deployment. You can access feature health status information using the splunkd health report in Splunk Web, or access feature health status information from the /server/health/splunkd endpoint.

For more information, see About proactive Splunk component monitoring.
About the Monitoring Console

About the Monitoring Console

What is the Monitoring Console?

The Monitoring Console is the Splunk Enterprise monitoring tool. It lets you view detailed topology and performance information about your Splunk Enterprise deployment. Before Splunk Enterprise version 6.5.0, the Monitoring Console was called the Distributed Management Console.

The available dashboards provide insight into the following areas of your deployment or instance:

- search performance and distributed search framework
- indexing performance
- operating system resource usage
- Splunk app key value store performance
- search head and indexer clustering
- index and volume usage
- forwarder connections and Splunk TCP performance
- HTTP Event Collector performance
- and license usage.

The Monitoring Console dashboards use data from Splunk Enterprise's internal log files such as metrics.log, as well as data available from Splunk Enterprise platform instrumentation.

Find the Monitoring Console

From anywhere in Splunk Web, click Settings, and then click the Monitoring Console icon on the left.
The Monitoring Console is visible only to users with the Splunk Enterprise admin role.

For information about the Monitoring Console in Splunk Cloud, see Monitor your Splunk Cloud Deployment in the Splunk Cloud Admin Manual.

What can the Monitoring Console do?

There are three main configuration states for the Monitoring Console.

- You can leave the Monitoring Console unconfigured in standalone mode on your Splunk Enterprise instance. This means that you can navigate to the Monitoring Console on your individual instance in your deployment and see that particular instance’s performance.
- You can go through the configuration steps, still in standalone mode, which lets you access the default platform alerts.
- You can go through the configuration steps for distributed mode, which lets you log into one instance and view the console’s information for every instance in your deployment.

Find answers to common problems

The Monitoring Console is a rich source of troubleshooting information about your Splunk Enterprise deployment. Here are a few examples of problems you can investigate with this tool.

<table>
<thead>
<tr>
<th>Symptom</th>
<th>Dashboard</th>
</tr>
</thead>
</table>
| My users are reporting errors when they run a search, like "Peer x is unresponsive", "Peer did not participate in search", or "Results might be incomplete". | Possible starting points include:  
  - Distributed search: Deployment health checks, or Distributed search: Instance if you know which search peer is experiencing the problems.  
  - On Resource usage: Deployment select search peers, and look for any that are oversubscribed.  
  - Compare problematic time periods in the Distributed search and Resource usage views to consider capacity planning. |
| My users’ UI is slow | Resource usage: Instance for the instance experiencing problems. |
| My search performance is slow | Resource usage: Deployment, Scheduler activity, or Search activity |
| Is my indexer/search head up/down right now? | Overview > Topology |
| Is indexer workload distributed evenly across instances? | Indexing performance: Deployment |
| What are my indexes’ retention policies? | Indexing > Indexes and Volumes: Instance |
| KV store is not initializing | Search > KV Store: Deployment |
| Splunk Web errors that disk space is full | Resource usage: Machine or Indexes and volumes dashboards |

How the Monitoring Console works

This topic lists the files that the Monitoring Console modifies in a Splunk Enterprise filesystem.

These files reside in $SPLUNK_HOME/etc/apps/splunk_monitoring_console/ unless indicated otherwise. This directory contains configuration files in both a default directory and, after Monitoring Console setup, a local directory. See About
configuration file directories in the *Admin Manual*.

<table>
<thead>
<tr>
<th>File(s)</th>
<th>Information contained in file(s)</th>
<th>When populated</th>
</tr>
</thead>
<tbody>
<tr>
<td>app.conf</td>
<td>Basic information about the Monitoring Console: determines whether it is in distributed mode, and provides a short description for Splunk Web to use in Launcher. See app.conf.spec.</td>
<td>By default. Updated when you click <em>Apply changes</em>.</td>
</tr>
<tr>
<td>distsearch.conf in etc/system/local</td>
<td>Contains stanzas that reference distributed search groups created by the Monitoring Console. The names of these groups are usually prefaced with dmc_group_. For example: [distributedSearch:dmc_group_cluster_master]</td>
<td>When you switch to distributed mode in Monitoring Console setup and click <em>Apply changes</em>.</td>
</tr>
<tr>
<td>dmc_alerts.conf</td>
<td>In some cases, you can edit thresholds in a platform alert without having to directly modify the search string for that alert. For such an alert, the Monitoring Console has a template of the search string, description string, and editable parameters. The template data, which is used in the Monitoring Console Alerts Setup page, is stored here, in stanzas named for the name of the saved search in default/savedsearches.conf.</td>
<td>By default</td>
</tr>
<tr>
<td>lookups directory</td>
<td>Contains two important files:</td>
<td>By default (on initial startup). Updated when you click <em>Apply changes</em> or Rebuild forwarder assets, respectively.</td>
</tr>
</tbody>
</table>
| assets.csv                      | • assets.csv lists the instances that the Monitoring Console recognizes and their peer URI (unique name), server name, host, machine (host fqdn), search group (server role, custom group, or cluster). This csv is used by every Monitoring Console dashboard.  
  • dmc_forwarder_assets.csv is generated when you enable forwarder monitoring. Enabling forwarder monitoring enables the scheduled search (DMC Forwarder - Build Asset Table) in savedsearches.conf, which populates this .csv file. See Configure forwarder monitoring for the Monitoring Console in this manual. |                                                                                   |
| macros.conf                     | Contains two types of macros:                                                                  | Search macros are stored here by default.                                      |
|                                | • Search macros for all Monitoring Console dashboards.                                        | Customizations are set when you edit one and click *Save*.                    |
|                                | • Overview page customizations set in Monitoring Console > Settings > Overview preferences.     |                                                                                  |
|                                | See macros.conf.spec.                                                                          |                                                                                  |
| props.conf                      | Search-time field extraction and lookup applications and evals. See props.conf.spec.           | By default                                                                      |
| savedsearches.conf             | Schedules and search strings for platform alerts. The saved search named DMC Forwarder - Build Asset Table runs when you enable forwarder monitoring. | By default                                                                      |
| splunk_monitoring_console_assets.conf | This file contains:  
  • A list of search peers configured with the Monitoring Console, and any for which you have disabled monitoring.  
  • Any search peer identifier that has been overwritten by the Monitoring Console manually during setup, for example host, host_fqdn, indexer cluster labels, or search head cluster labels.  
  • Stanzas describing which indexer and search head cluster(s) each search peer is a member of. | When you click “Apply Changes” on Setup > General setup                           |
| transforms.conf                | Lookup definitions for assets.csv and forwarder csv file                                      | By default                                                                      |
Configure the Monitoring Console

Multi-instance Monitoring Console setup steps

This topic is an overview of the steps required to set up your monitoring console for a distributed Splunk Enterprise deployment. To configure the monitoring console on a standalone instance, see Single-instance Monitoring Console setup steps in this manual.

To set up a monitoring console for a distributed deployment, perform the following steps:

<table>
<thead>
<tr>
<th>Step number</th>
<th>Description of step</th>
<th>How to proceed with this step</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Decide which instance will host the monitoring console for your deployment.</td>
<td>See Which instance should host the console?</td>
</tr>
<tr>
<td>2</td>
<td>Ensure your deployment meets the prerequisites.</td>
<td>See Monitoring console setup prerequisites.</td>
</tr>
<tr>
<td>3</td>
<td>Set search head cluster and indexer cluster labels.</td>
<td>See Set cluster labels.</td>
</tr>
<tr>
<td>4</td>
<td>Add all instances as search peers.</td>
<td>See Add instances as search peers to the monitoring console.</td>
</tr>
<tr>
<td>5</td>
<td>Set up the monitoring console in distributed mode.</td>
<td>See Configure the monitoring console in distributed mode.</td>
</tr>
<tr>
<td>6 (optional)</td>
<td>Use the monitoring console forwarder dashboards.</td>
<td>See Configure forwarder monitoring for the Monitoring Console.</td>
</tr>
<tr>
<td>7 (optional)</td>
<td>Enable platform alerts.</td>
<td>See Enable and configure platform alerts.</td>
</tr>
<tr>
<td>8 (optional)</td>
<td>Modify or add Health Check items.</td>
<td>See Customize Health Check.</td>
</tr>
<tr>
<td>9 (optional)</td>
<td>Customize color mappings for the Overview page.</td>
<td>Navigate to Monitoring Console &gt; Settings &gt; Overview Preferences.</td>
</tr>
</tbody>
</table>

Get started

To set up your monitoring console for a distributed deployment, first decide where to host it in your deployment. Begin with Which instance should host the console?

Single-instance Monitoring Console setup steps

This topic is an overview of the steps required to set up your monitoring console for a standalone Splunk Enterprise instance. To configure the monitoring console for a distributed Splunk Enterprise deployment, see "Multi-instance monitoring console setup steps."

To set up a monitoring console for a standalone deployment, perform the following steps:

<table>
<thead>
<tr>
<th>Step number</th>
<th>Description of step</th>
<th>How to proceed with this step</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Ensure your deployment meets the prerequisites.</td>
<td>See Monitoring console setup prerequisites.</td>
</tr>
<tr>
<td>2</td>
<td>Set up the monitoring console in standalone mode.</td>
<td>See Configure Monitoring Console in standalone mode.</td>
</tr>
<tr>
<td>3 (optional)</td>
<td>Use the monitoring console forwarder dashboards.</td>
<td>See Configure forwarder monitoring for the Monitoring Console.</td>
</tr>
</tbody>
</table>
To configure your monitoring console for standalone deployment, first verify the prerequisites on your instance. Begin with Monitoring Console setup prerequisites.

**Which instance should host the console?**

This topic is a step in the process of setting up the monitoring console for a distributed Splunk Enterprise deployment.

To start, determine which instance will best host the monitoring console. You have several options for where to host the monitoring console, depending on the nature of your deployment:

- The instance you choose must meet or exceed the search head reference hardware requirements. See Reference hardware in the *Capacity Planning Manual*.
- For security and performance reasons, only Splunk Enterprise administrators should have access to this instance.
- The instance hosting the monitoring console must not run any searches unrelated to its function as monitoring console. The exception to this rule is if you are using the console to monitor a standalone single-instance deployment.

This table outlines the recommended locations for the monitoring console, based on deployment type.

<table>
<thead>
<tr>
<th>Distributed mode?</th>
<th>Indexer clustering?</th>
<th>Search head clustering?</th>
<th>Recommended locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>No</td>
<td>N/A</td>
<td>N/A</td>
<td>The standalone instance.</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>The license master or a deployment server servicing a small number (&lt;50) of clients. Otherwise, run the monitoring console on a search head that is dedicated to running monitoring console searches.</td>
</tr>
<tr>
<td>Yes</td>
<td>Single indexer cluster</td>
<td>Not relevant</td>
<td>The master node, if the load on the master node is below the limits specified in Additional roles for the master node in the <em>Managing Indexers and Clusters of Indexers</em> manual. Otherwise, run the monitoring console on a search head node that is dedicated to running monitoring console searches. If you are using SmartStore you must host the monitoring console on a dedicated search head.</td>
</tr>
<tr>
<td>Yes</td>
<td>Multiple indexer clusters</td>
<td>Not relevant</td>
<td>A search head that is configured as a search head node across all the clusters. This search head must be dedicated to monitoring console use.</td>
</tr>
<tr>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>The search head cluster deployer, a license master, or a standalone search head that is dedicated to running monitoring console searches. Do not run the monitoring console on a search head cluster member.</td>
</tr>
</tbody>
</table>

For a general discussion of management component colocation, see Components that help to manage your deployment in the *Distributed Deployment Manual*.

See the sections that follow for detailed information for certain deployment types.
In a non-clustered deployment

You can locate the monitoring console on any of these instances:

- A license master
- A deployment server that is servicing a small number (<50) of clients
- A dedicated search head

In a deployment with a single indexer cluster

In a single indexer cluster, you can host the monitoring console on the instance running the master node if the load on the master node is below the limits specified in Additional roles for the master node in the Managing Indexers and Clusters of Indexers manual.

You can also host the monitoring console on a search head node in the cluster, but you must dedicate the node to monitoring console searches. You cannot use the search head to run any other searches.

If you are using SmartStore, you must host the monitoring console on a dedicated search head.

In a deployment with multiple indexer clusters

If your deployment has multiple indexer clusters, host the monitoring console on a dedicated search head configured as a search head node on each indexer cluster. Do not use this search head to run any non-monitoring console searches.

To do this:

1. Configure a search head to serve as a node on each of the indexer clusters. See Search across multiple indexer clusters in the Managing Indexers and Clusters of Indexers manual. This is your monitoring console instance.

2. Configure each master node and all search head nodes in the clusters as search peers of the monitoring console instance. See Add instances as search peers in this manual.

Do not configure the cluster peer nodes (indexers) as search peers to the monitoring console node. As nodes in the indexer clusters, they are already known to all search head nodes in their cluster, including the monitoring console node.

In a deployment with a search head cluster but without an indexer cluster

You can locate the monitoring console on any of these instances:

- A search head cluster deployer
- A license master
- A standalone, dedicated search head

Do not run the monitoring console on a search head cluster member.

The Monitoring Console is not supported for search head pooling deployments. Search head pooling was first deprecated in Splunk Enterprise 6.2.0 and the functionality is removed from Splunk Enterprise 8.0.0 and higher.
Why not to host the console on a production search head

Do not configure the monitoring console on an existing production search head that is already in use for the following reasons:

- Non-monitoring console searches that run on this search head might have incomplete results. The monitoring console distributed search groups modify default search behavior to ensure that the searches for the monitoring console dashboards are narrowly scoped to the list of search peers that they target. When you set up the monitoring console in distributed mode, it creates one search group for each server role, identified cluster, or custom group. Unless you use a "splunk_server_group" or the "splunk_server" option, only search peers that are members of the indexer group are searched by default. Because all searches that run on the monitoring console instance follow this behavior, non-monitoring console searches might have incomplete results.
- All production search heads should be monitored for performance, and the monitoring console affects the performance of the search head that hosts it. It can be difficult to disentangle monitoring console resource usage from production resource usage on the same instance.

The monitoring console and deployment server

In most cases, you cannot host the distributed monitoring console on a deployment server. The exception is if the deployment server handles only a small number of deployment clients, no more than 50. The monitoring console and deployment server functionalities can interfere with each other at larger client counts. See Deployment server provisioning in the Updating Splunk Enterprise Instances manual.

Next step

To continue setting up the monitoring console in distributed mode, make sure your deployment meets the prerequisites. See Monitoring Console setup prerequisites.

Monitoring Console setup prerequisites

This topic is a step in the process of setting up the monitoring console for either a distributed Splunk Enterprise deployment or a standalone Splunk Enterprise instance.

To proceed with your monitoring console deployment, verify that you meet the following setup prerequisites:

- Have a functional Splunk Enterprise deployment.
- Ensure that each instance in the deployment has a unique server.conf serverName value and inputs.conf host value.
- Enable platform instrumentation for every Splunk Enterprise instance that you intend to monitor, except forwarders. See About Splunk Enterprise platform instrumentation.
- Forward internal logs (both $SPLUNK_HOME/var/log/splunk and $SPLUNK_HOME/var/log/introspection) to indexers from all other components. Without this step, many dashboards will lack data. See Best practice: Forward search head data in the Distributed Search Manual.
- The user setting up the monitoring console needs the admin_all_objects capability.
Dashboard version dependencies

The dashboards in the monitoring console rely on data collected from Splunk Enterprise internal log files and endpoints. Much of the data comes from platform instrumentation, which was introduced in Splunk Enterprise version 6.1 and enhanced in subsequent releases. The following table summarizes the minimum Splunk Enterprise version requirements for particular platform instrumentation capabilities. If the instances that you monitor in the console do not meet these minimum version requirements, the related dashboard panels are empty.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Panel</th>
<th>Minimum version requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>All dashboards</td>
<td>Most panels</td>
<td>Splunk Enterprise 6.1</td>
</tr>
<tr>
<td>KV store dashboards</td>
<td>All panels</td>
<td>Splunk Enterprise 6.2.0 (which introduced the KV store)</td>
</tr>
<tr>
<td>Search head clustering dashboards</td>
<td>All panels</td>
<td>Splunk Enterprise 6.2.0</td>
</tr>
<tr>
<td>Distributed search dashboards</td>
<td>Panels about bundle replication</td>
<td>Splunk Enterprise 6.3.0</td>
</tr>
<tr>
<td>HTTP Event Collector (HEC) dashboards</td>
<td>All panels</td>
<td>Splunk Enterprise 6.3.0 (which introduced HEC)</td>
</tr>
<tr>
<td>Scheduler dashboards</td>
<td>Most panels</td>
<td>Splunk Enterprise 6.3.0</td>
</tr>
<tr>
<td>Resource usage: Machine, Resource usage:</td>
<td>I/O panels</td>
<td>Splunk Enterprise 6.4.0</td>
</tr>
<tr>
<td>Deployment</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Health check</td>
<td>N/A</td>
<td>Splunk Enterprise 6.5.0 on instance hosting monitoring console</td>
</tr>
</tbody>
</table>

**Next step**

To continue setting up your monitoring console in a distributed deployment, see Set cluster labels.

To continue setting up your monitoring console on a standalone instance, skip to Configure Monitoring Console in standalone mode.

**Set cluster labels**

This topic is a step in the process of setting up the monitoring console for a Splunk Enterprise deployment.

Set labels on your indexer clusters and search head clusters, so that the monitoring console can identify the instances associated with them. The labels allow the monitoring console to populate the indexer clustering and search head clustering dashboards.

**During cluster deployment**

Set the label during initial deployment of the cluster. If necessary, you can set or change the label later.

To set the indexer cluster label during deployment, see Enable the indexer cluster master node in the Managing Indexers and Clusters of Indexers manual.

To set the search head cluster label during deployment, see Deploy a search head cluster in the Distributed Search manual.
After cluster deployment

If you are deploying a new cluster, set cluster labels during the deployment process. If your cluster is already deployed, set cluster labels at the beginning of the monitoring console setup according to the following tables.

<table>
<thead>
<tr>
<th>Splunk Enterprise version and cluster type</th>
<th>Where?</th>
<th>How?</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.3.0+ indexer cluster</td>
<td>On the cluster master</td>
<td>See Configure the master with the dashboard in the Managing Indexers and Clusters of Indexers manual. Alternatively, run the following CLI command from the cluster master: splunk edit cluster-config -cluster_label &lt;CLUSTER LABEL&gt;</td>
</tr>
<tr>
<td>Pre-6.3.0 indexer cluster</td>
<td>On the monitoring console setup page</td>
<td>Set label for all instances in cluster</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Splunk Enterprise version and cluster type</th>
<th>Where?</th>
<th>How?</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.3.0+ search head cluster members</td>
<td>On any search head cluster member</td>
<td>See Configure the search head cluster in the Distributed Search manual. Alternatively, run the following CLI command from any cluster member: splunk edit shcluster-config -shcluster_label &lt;CLUSTER LABEL&gt;</td>
</tr>
<tr>
<td>Pre-6.3.0 search head cluster members</td>
<td>On the monitoring console setup page</td>
<td>Set the label for all instances in the cluster</td>
</tr>
<tr>
<td>Search head cluster deployer</td>
<td>On the monitoring console setup page</td>
<td>Set the label for the deployer instance</td>
</tr>
</tbody>
</table>

Whichever method you use to edit cluster labels, go to the monitoring console setup page and click Apply Changes to update the monitoring console asset table.

About the monitoring console setup page

The monitoring console setup page in Splunk Web is at Monitoring Console > Settings > General Setup.

After editing cluster labels, click Apply Changes.

Next step

To continue setting up your monitoring console in a distributed deployment, see Add instances as search peers to the Monitoring Console.

Add Splunk Enterprise instances to the Monitoring Console

This topic is a step in the process of setting up the monitoring console for a distributed Splunk Enterprise deployment.

The monitoring console is a Splunk Enterprise instance that functions as a search head. It runs searches across other instances to gather information from those instances. However, for it to gather information, you must add those instances
as search peers of the search head.

You must add each instance that you want to monitor to the monitoring console as a search peer, regardless of the server role, with the exception of indexers that are part of an indexer cluster.

To add Splunk Enterprise instances as search peers to the monitoring console:

1. Log into the instance on which you want to configure the monitoring console.
2. In Splunk Web, click Settings > Distributed search > Search peers.
3. Click New.
4. Fill in the requested fields, and click Save.
5. Repeat steps 3 and 4 for each search head, deployment server, license master, non-clustered indexer, and clustered search head. Do not add clustered indexers. If you are monitoring an indexer cluster and you are hosting the monitoring console on an instance other than the cluster master, you must add the cluster master as a search peer and you must configure the monitoring console instance as a search-head in that cluster.

Next step

To continue setting up the monitoring console on a distributed deployment, see Configure Monitoring Console in distributed mode.

Configure Monitoring Console in standalone mode

This topic is a step in the process of setting up the monitoring console for a standalone Splunk Enterprise instance.

To configure the monitoring console for a standalone instance:

1. In Splunk Web, navigate to Monitoring Console > Settings > General Setup.
2. Check that search head, license master, and indexer are listed under Server Roles, and nothing else. If not, click Edit to correct.
3. Click Apply Changes.

Next step

To continue setting up the monitoring console for a standalone deployment, skip to Configure forwarder monitoring for the Monitoring Console.

Configure the Monitoring Console in distributed mode

This topic is a step in the process of setting up the monitoring console for a distributed Splunk Enterprise deployment.

To configure the monitoring console in distributed mode:

1. Log into the instance on which you want to configure the monitoring console. The instance by default is in standalone mode, unconfigured.
2. In Splunk Web, select Monitoring Console > Settings > General Setup.
3. Click Distributed mode.
4. Confirm the following:
   ♦ The columns labeled instance and machine are populated correctly and show unique values within each column.
   ♦ The server roles are correct. For example, a search head that is also a license master must have both server roles listed. If not, click Edit > Edit Server Roles and select the correct server roles for the instance.
   ♦ If you are using indexer clustering, make sure the cluster master instance is set to the cluster master server role. If not, click Edit > Edit Server Roles and select the correct server role.
   ♦ If you are hosting the monitoring console on an instance other than the cluster master, you must add the cluster master instance as a search peer and configure the monitoring console instance as a search head in that cluster. See Add Splunk Enterprise instances to the Monitoring Console.
   ♦ Make sure anything marked as an indexer is actually an indexer.
5. (Optional) Set custom groups. Custom groups are tags that map directly to distributed search groups. You might find groups useful, for example, if you have multisite indexer clustering in which each group can consist of the indexers in one location, or if you have an indexer cluster plus standalone peers. Custom groups are allowed to overlap. For example, one indexer can belong to multiple groups. See Create distributed search groups in the Distributed Search manual.
6. Click Apply Changes.

If you add another node to your deployment later, click Settings > General Setup and check that these items are accurate.

Reset server roles after restart

After you configure the monitoring console in distributed mode, a restart of the instance hosting the monitoring console can cause any changes to server role settings on the instances you are monitoring to be lost.

To properly reset server roles after restart of the distributed monitoring console:

1. Click Settings > General Setup > Reset All Settings > Refresh. This restores the monitoring console to its original default configuration.
2. Click Distributed.
3. For the remote instances whose server roles you want to change, click Edit.
4. Select or remove the specific server roles. Click Save.
5. Click Apply Changes.

Next step

To configure the monitoring console for forwarders, see Configure forwarder monitoring for the Monitoring Console.

Configure forwarder monitoring for the Monitoring Console

This topic is a step in the process of setting up the monitoring console for either a distributed or a standalone Splunk Enterprise deployment.

Prerequisites

For several dashboard monitoring panels to work, your forwarders need unique and persistent GUIDs. One way to accomplish this is to clone your forwarder before starting it. The forwarder GUID is in instance.cfg, which populates when you start the forwarder.
Setup

In Splunk Web, click Monitoring Console > Settings > Forwarder Monitoring setup and follow the setup steps.

About time settings

On the forwarder monitoring setup page, you can enable or disable forwarder monitoring and set the data collection interval. Enabling forwarder monitoring runs a scheduled search that populates dmc_forwarder_assets.csv, a lookup file that resides on the monitoring console node in $SPLUNK_HOME/etc/apps/splunk_monitoring_console/lookups. The monitoring console uses this forwarder asset table to know which forwarders to display information about in the forwarder monitoring dashboards.

In Splunk Web click Settings > Searches and reports > DMC Forwarder - Build Asset Table to review the scheduled search.

Click Monitoring Console > Settings > Forwarder Monitoring Setup and choose from several values for data collection interval. This interval determines how often that scheduled search runs. The default value is 15 minutes.

When the scheduled search runs to rebuild the forwarder asset table it always looks back 15 minutes. This lookback time is not configurable, and it is different from the data collection interval. For example, if you set the data collection interval to 24 hours, the scheduled search will run once every 24 hours, but check only the 15 minutes before it starts running.

Scheduled search can be expensive if you have many forwarders. You might want to run the search less often than the default value.

Rebuild the forwarder asset table

The data in the forwarder asset table is cumulative. If a forwarder connects to an indexer, its record exists in the table. If you later remove the forwarder from your deployment, the forwarder's record is not removed from the asset table. It is instead marked "missing" in the asset table, and it still appears in the DMC forwarder dashboards.

To remove a forwarder entirely from the monitoring console dashboards, click rebuild forwarder assets in Monitoring Console > Settings > Forwarder Monitoring Setup. You can choose a lookback time when you perform this action. The lookback selection during this action does not change the 15-minute lookback time for the scheduled search or the data collection interval discussed elsewhere in this topic.

Next step

To set up platform alerts, see Enable and configure platform alerts. This step is optional.

Enable and configure platform alerts

Platform alerts are saved searches included in the monitoring console. Platform alerts notify Splunk Enterprise administrators of conditions that might compromise their deployment environment. When an alert is triggered, the Monitoring Console Overview page displays a notification. You can also view the alert and its results by going to Overview > Alerts > Managed triggered alerts.

The included platform alerts get their data from REST endpoints. Platform alerts are disabled by default.
Enable platform alerts

Prerequisite

Configure your Monitoring Console. See Single instance Monitoring Console setup steps or Multi-instance Monitoring Console setup steps depending on your deployment type.

1. From the monitoring console Overview, click Triggered Alerts > Enable or Disable.
2. Click the Enabled check box next to the alerts that you want to enable.

You can also set an alert action, such as an email notification.

Configure platform alerts and set alert actions

You can view and configure the default settings and parameters for platform alerts, including the following:

- Alert thresholds, if applicable
- Alert schedule
- Suppression time
- Alert actions (such as sending an email or starting a custom script)

To change an alert threshold, perform the following steps:

1. From the Monitoring Console, click Overview > Alerts > Enable or Disable.
2. Find the alert you want to configure and click Edit.
3. Edit the threshold field to your desired value.
4. Click Save.

To view and edit advanced settings like alert schedule, trigger conditions, and alert actions, perform the following steps:

1. From the Monitoring Console, click Overview > Alerts > Enable or Disable.
2. Find the alert you want to configure and click Advanced edit.
3. Modify the settings, if desired.
4. Click Save.

If you enable email notifications, make sure that you have defined a valid mail host in Settings > Server settings > Email settings.

For guidance on alert actions, see Set up alert actions in the Alerting Manual.

You can also view the complete list of default parameters for platform alerts in $SPLUNK_HOME/etc/apps/splunk_management_console/default/savedsearches.conf. If you choose to edit configuration files directly, put the new configurations in a local directory instead of the default.

Never edit configuration files in the default directory.

Default platform alerts in the monitoring console

The following platform alerts are available by default in the monitoring console. To monitor your deployment with platform alerts, enable the individual alerts that you want.
<table>
<thead>
<tr>
<th>Alert name</th>
<th>Description</th>
<th>For more information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abnormal State of Indexer Processor</td>
<td>Fires when one or more of your indexers reports an abnormal state. This abnormal state can be either throttled or stopped.</td>
<td>For details on which indexer is in which abnormal state, and to begin investigating causes, see the monitoring console <a href="#">Indexing Performance: Deployment</a> dashboard's <a href="#">Indexing Performance by Instance</a> panel. For information about the dashboard, see <a href="#">Indexing performance dashboards</a> in this manual.</td>
</tr>
<tr>
<td>Critical System Physical Memory Usage</td>
<td>Fires when one or more instances exceeds 90% memory usage (by any process, Splunk software or otherwise). On most Linux distributions, this alert can trigger if the OS is engaged in buffers and filesystem caching activities. The OS releases this memory if other processes need it, so it does not always indicate a serious problem.</td>
<td>For details on instance memory usage, navigate to the monitoring console <a href="#">Resource Usage: Deployment</a> dashboard. For information about the dashboard, see <a href="#">Resource usage dashboards</a> in this manual.</td>
</tr>
<tr>
<td>Expired and Soon To Expire Licenses</td>
<td>Fires when you have licenses that have expired or will expire within two weeks.</td>
<td>For information about your licenses and license usage, click <a href="#">Licensing</a> in the monitoring console.</td>
</tr>
<tr>
<td>Missing forwarders</td>
<td>Fires when one or more forwarders are missing.</td>
<td>See the forwarders dashboards in the monitoring console.</td>
</tr>
<tr>
<td>Near Critical Disk Usage</td>
<td>Fires when you have used 80% of your disk capacity.</td>
<td>For more information about your disk usage, navigate to the three monitoring console <a href="#">Resource Usage</a> dashboards and read <a href="#">Resource usage dashboards</a> in this manual.</td>
</tr>
<tr>
<td>Saturated Event-Processing Queues</td>
<td>Fires when one or more of your indexer queues reports a fill percentage, averaged over the last 15 minutes, of 90% or more. This alert can inform you of potential indexing latency.</td>
<td>For more details about your indexer queues, navigate to the two monitoring console <a href="#">Indexing Performance</a> dashboards and read <a href="#">Indexing performance dashboards</a> in this manual.</td>
</tr>
<tr>
<td>Search Peer Not Responding</td>
<td>Fires when any of your search peers (indexers) is unreachable.</td>
<td>For the status of all your instances, see the monitoring console <a href="#">Instances</a> view.</td>
</tr>
<tr>
<td>Total License Usage Near Daily Quota</td>
<td>Fires when you have used 90% of your total daily license quota.</td>
<td>For more information about your license usage, click <a href="#">Licensing</a> in the monitoring console.</td>
</tr>
</tbody>
</table>

### Longevity of platform alert search artifacts

In `savedsearches.conf`, the `dispatch.ttl` setting dictates that the searches from platform alerts keep search artifacts for four hours. But if an alert is triggered, its search artifact stays for seven days. This means that the link sent in an email to inspect the search results of a triggered alert expires in seven days (by default).

### Next step

To set up health checks, see [Access and customize health check](#). This step is optional.

### Access and customize health check

The monitoring console comes with preconfigured health checks. You can modify existing health checks and create or download new ones.
Use the health check

Find the health check at Monitoring Console > Health Check. Start the health check by clicking Start.

Each health check item runs a separate search. The searches run sequentially. When one search finishes, the next one starts. After all searches have completed, the results are sorted by severity: Error, Warning, Info, Success, or N/A.

Click a severity level at the top of the results to see only results with that severity level. Click a row to see more information, including suggested actions.

To run only some of the checks, filter by tag or category before clicking Start. From the monitoring console, you can run health checks that have been created in any app installed on your monitoring console node. Use the app drop-down list to filter health checks by app context.

Exclude a check

You can disable a specific check to prevent it from running when you click Start:

1. Click Monitoring Console > Settings > Health Check Items.
2. Locate the check you wish to disable in the list.
3. Click Disable.
4. Reload Monitoring Console > Health Check.

You can also filter the checks by group, app, tag, and category at the top of the page before clicking Start.

Modify an existing check

You can modify an existing check. For example, to modify the warning threshold for the Excessive physical memory usage check from 90% to 80%:

1. Click Monitoring Console > Settings > Health Check Items.
2. In the Excessive physical memory usage row, click Edit.
3. Edit the Search and Description fields.
4. (Optional) Rename the health check item to reflect your modification.
5. Click Save.

The modifications are saved to your filesystem in $SPLUNK_HOME/etc/apps/splunk_monitoring_console/local/checklist.conf

Create a new health check

You can add a new health check item as follows:

1. Click Monitoring Console > Settings > Health Check Items.
2. Click New Health Check Item.
3. Fill in the title and ID fields.
4. (Optional) Choose an app context for this check. The default is monitoring console.
5. Continue filling in the fields. Be sure to include a severity level in your search (| eval severity_level). Without this, the search returns results as N/A. See About searches for guidance filling in the Search field.
6. (Optional) For Environments to exclude, select either Standalone or Distributed. Any other value in this field is ignored. See What can the monitoring console do? for information about standalone and distributed modes.
7. Click **Save**.

The modifications are saved to your filesystem in `${SPLUNK_HOME}/etc/apps/<app_name>/local/checklist.conf` on *nix or `${%SPLUNK_HOME%}/etc/apps/<app_name>/local/checklist.conf` on Windows. If you do not specify an app context, the modifications are saved in the `splunk_monitoring_console` app directory.

**Search results format**

In standalone mode, the search string generates the final result. In distributed mode, this search generates one row per instance in the result table.

The search results must be in the following format.

- **instance metric severity_level**
  - `<instance name>` `<metric number or string>` `<level number>`

Severity level names correspond to values as follows.

<table>
<thead>
<tr>
<th>Severity level name</th>
<th>Severity level value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error</td>
<td>3</td>
</tr>
<tr>
<td>Warning</td>
<td>2</td>
</tr>
<tr>
<td>Info</td>
<td>1</td>
</tr>
<tr>
<td>Success</td>
<td>0</td>
</tr>
<tr>
<td>N/A</td>
<td>-1</td>
</tr>
</tbody>
</table>

**Add a drilldown to a search or dashboard**

You can also include a drilldown to another search or to a dashboard, for example a monitoring console dashboard, in your health check results.

To include a monitoring console dashboard drilldown:

1. Choose an existing dashboard in the monitoring console that is relevant to the data you want to run a health check on. Choose a dashboard that has a drop-down list to choose an instance or machine.
2. Inspect the URL using the drop-down list to see which parts of the URL are needed to specify the instance you want. Look for `&form.splunk_server=$instance$` toward the end of the URL.
3. Trim the URL to a URI that starts with `/app/` and has a `$` delimited variable name that is a column in the search results for your health check item. For example,

   `/app/splunk_monitoring_console/distributed_search_instance?form.splunk_server=$search_head$`

To include a search drilldown, find or create a search with a `$` delimited variable in it. The variable must exist as a column name in the health check search results. For example, a drilldown of `index=_internal $instance$` will work, as long as "instance" is a column name in the health check search.

Most likely, you want a drilldown search of the search you just ran. In that case, replace `$rest_scope$` or `$hist_scope$` with `$instance$`, where instance is a column name in the health check search. For example:

```
dmc_set_index_internal' host=$instance$ earliest=-60m source=*splunkd.log* (component=AggregatorMiningProcessor OR component=LineBreakingProcessor OR component=DateParserVerbose) (log_level=WARN OR log_level=ERROR)
```
**Proactively alert on health check conditions**

Many health check items already have a corresponding platform alert. You can also turn an additional health check into an alert.

This table lists the health check items with corresponding platform alerts:

<table>
<thead>
<tr>
<th>Health check</th>
<th>Corresponding platform alert</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indexing status</td>
<td>Abnormal State of Indexer Processor</td>
<td>Tests the current status of the indexer processor on indexer instances.</td>
</tr>
<tr>
<td>Excessive physical memory usage</td>
<td>Critical System Physical Memory Usage</td>
<td>Assesses system-wide physical memory usage and raises a warning for those servers where it is &gt;90%.</td>
</tr>
<tr>
<td>Expiring or expired licenses</td>
<td>Expired and Soon To Expire Licenses</td>
<td>Checks for licenses that are expired or will expire within 2 weeks.</td>
</tr>
<tr>
<td>Missing forwarders</td>
<td>Missing forwarders</td>
<td>Checks for forwarders that have not connected to indexers for &gt;15 minutes in the recent past.</td>
</tr>
<tr>
<td>Near-critical disk usage</td>
<td>Near Critical Disk Usage</td>
<td>Checks for 80% of the disk usage of partitions that Splunk Enterprise reads or writes to.</td>
</tr>
<tr>
<td>Saturation of event-processing queues</td>
<td>Saturated Event-Processing Queues</td>
<td>One or more of your indexer queues is reporting a fill percentage, averaged over the last 15 minutes, of 90% or more.</td>
</tr>
<tr>
<td>Distributed search health assessment</td>
<td>Search Peer Not Responding</td>
<td>Checks the status of the search peers (indexers) of each search head.</td>
</tr>
</tbody>
</table>

To create a new alert from a health check when a counterpart does not already exist:

1. Run the health check.
2. Click **Open in search**.
3. Modify the search with a **where** clause.
4. Save it as a new scheduled search with an alert action. For example, email the admin.

**Export health check results**

You can export the results from a health check item to your local machine to share with others.

To export results from a health check item:

1. Run the health check.
2. Click the row with the results you want to export.
3. In the results table on the right, click **Export**.
4. Choose the format of the results (XML, CSV, or JSON). You can also choose a file name and the number of results.
5. Click **Export**.
Monitoring Console dashboard reference

Indexing: Performance

This topic is a reference for the Performance dashboards under the Indexing menu in the Monitoring Console.

For more information about troubleshooting indexing performance problems, see Identify and triage indexing performance problems in the Troubleshooting Manual.

Indexing Performance: Deployment

The Indexing Performance: Deployment dashboard provides an overview of indexing performance across your Splunk Enterprise deployment.

In the Overview of Indexing Performance panel, total indexing rate is summed over all indexers.

In the Instances by Estimated Indexing Rate panel, the indexing rate is estimated because it uses metrics.log, which takes only the top ten results for each type by default. See About metrics.log in the Troubleshooting Manual.

Indexing Performance: Instance

The Indexing Performance: Instance dashboard contains panels about indexing performance on a per-indexer basis. These panels can provide more detailed insight into problems that surface in the deployment-wide dashboard.

The Splunk Enterprise Data Pipeline panel exposes decaying averages for queue sizes. The averages use data over the previous 15 minutes. This panel, along with the historical panel Median Fill Ratio of Data Processing Queues, helps you narrow down sources of indexing latency to a specific queue. Data starts at parsing and travels through the data pipeline to indexing at the end.

Here is an example of the Splunk Enterprise Data Pipeline panel for an instance with unhealthy queues:
In this example, although the parsing and aggregator queues have very high fill ratios, the problem is likely to be with processes in the typing queue. The typing queue is the first one that slows down, and data is backing up into the other two queues while waiting to get into the typing queue.

**Indexing Performance: Advanced**

The primary purpose of the Indexing Performance: Advanced dashboard is to provide information about pipeline set performance. You can use the dashboard to gain insight into the activity of the pipeline sets and their component pipelines. See Manage pipeline sets for index parallelization in Managing Indexers and Clusters of Indexers.

This dashboard is primarily of use when troubleshooting performance issues in consultation with Splunk Support. Without expert-level knowledge of the underlying processes, it can be difficult to interpret the information sufficiently to determine performance issue remediation.

The dashboard also includes panels with information on a few other advanced performance metrics, such as CPU usage. One of these panels, the **Aggregate CPU Seconds Spent per Indexer Processor Activity** panel, lets you “Split index service by subtask.” The several index services are subtasks related to preparing for and cleaning up after indexing. For more information about the subtask categories, see the metrics.log topic in the Troubleshooting Manual.

**Troubleshoot these dashboards**

The snapshot panels get data from Splunk REST endpoints for introspection. If snapshot panels lack data, check

- the system requirements for platform instrumentation.
- the `pipelinesets` setting in `server.conf`. When pipeline sets are used (that is, if `pipelinesets` is set to a value greater than 1), some panels of the Monitoring Console indexing performance dashboards will be blank.

The historical panels for these dashboards get data from metrics.log.

**Indexing: Indexes and Volumes**

This topic is a reference for all of the Indexing: Indexes and volumes dashboards in the Monitoring Console. See "About the Monitoring Console."

**What do these views show?**

The indexes and volume dashboards consist of six individual dashboards about indexing, each consisting of several panels.

Altogether the indexes and volumes dashboards describe how disk is being used on your indexes. These views break down the data presented in the resource usage views.

**Interpret results in these views**

If your system is tight on storage, visit the indexes and volumes dashboards. Use this data to help you evaluate and revise your retention policy. See "How the indexer stores indexes" in Managing Indexers and Clusters of Indexers.
What to look for in these views

In the **Indexes and Volumes: Instance** dashboard, instances highlighted in blue are rolling buckets to frozen. See "How the indexer stores indexes" in *Managing Indexers and Clusters of Indexers*.

In the **Indexes and Volumes: Deployment** dashboard, information about data that is being frozen can be a red flag. The **Indexes** and **Volumes** panels display this information.

In the **Index Detail: Deployment** dashboard, check the **Instances** panel for indexers with Data Age vs. Frozen Age where the data is being frozen well before their age limit. Drilldowns in the dashboard help you investigate.

---

**Indexing: Inputs: HTTP Event Collector**

This topic is a reference for the **HTTP Event Collector** dashboards in the Monitoring Console. See [About the Monitoring Console](#).

What do these views show?

The Monitoring Console provides two dashboards for the HTTP Event Collector, one with an overview of the performance across your deployment, and one with details about performance on each instance within your deployment.

What should I look for?

You can start in the deployment-wide dashboard to survey for outlying performance. Drill down to find more details on a problem in the related instance-scoped dashboard.

Interpret results in these views

In the **HTTP Event Collector: Deployment** dashboard, the list of tokens shown is a list of currently configured tokens. So if your token has been disabled, it will not appear in the **Token** dropdown. In contrast, the instance-scoped view shows all tokens, enabled and disabled.

See [Set up and use HTTP Event Collector in *Getting Data In*](#).

Troubleshoot these views

The instances doing the collecting must run Splunk Enterprise 6.3.0 or greater to use HTTP Event Collector. If your instance is a universal forwarder, the Monitoring Console cannot monitor it, because it cannot be a search peer.

If these dashboards lack data, verify that you have completed all the setup steps for the Monitoring Console, in either **distributed** or **standalone** mode.

---

**Indexing: Inputs: Data Quality**

This topic is a reference for the **Data Quality** dashboard in the Monitoring Console. See [About the Monitoring Console](#).
What does this dashboard show?

This dashboard report issues with event processing, such as:

- automatic source typing
- line breaking
- time stamp extraction
- time zone detection
- line merging
- excessively large events (high line count and/or large event size, \texttt{len(_raw)})
- indexing latency (\texttt{indextime - time})

Interpret results in this dashboard

See Resolve data quality issues in \textit{Getting Data In}.

What to look for in this dashboard

The \textbf{Event processing issues by source type} panel contains a table. If the table shows numbers in any of the cells, click for more information and to get started resolving issues.

Troubleshoot this dashboard

This dashboard uses data from \texttt{splunkd.log}.

If the drilldowns are loading slowly, you might have more issues than the drilldown searches can handle reasonably. Try narrowing the time range at the top of the page.

Indexing: License Usage

The License Usage reports in the Monitoring Console present the same information as the License Usage reports available on the License Master instance. See About the Splunk Enterprise license usage report view in the \textit{Admin Manual}.

If your Splunk Enterprise deployment has multiple license masters, you can use the License Usage reports on the Monitoring Console to select and view license reports for each license master.

Indexing: SmartStore

Several dashboards monitor \textbf{SmartStore} status. The dashboards are scoped either to a single instance or to the entire deployment. Find the dashboards under the \textbf{Indexing} menu and the \textbf{SmartStore} submenu:

- SmartStore Activity: Instance
- SmartStore Activity: Deployment
- SmartStore Cache Performance: Instance
- SmartStore Cache Performance: Deployment
SmartStore Activity dashboards

The SmartStore Activity dashboards provide information on activity related to the remote storage, such as:

- Remote storage connectivity
- Bucket upload/download activity
- Bucket upload/download failure count

The SmartStore Activity dashboards also include check boxes that you can select to show progress if you are currently performing data migration or bootstrapping.

SmartStore Cache Performance dashboards

The SmartStore Cache Performance dashboards provide information on the local caches, such as:

- The values for the `server.conf` settings that affect cache eviction
- The bucket eviction rate
- Portion of search time spent downloading buckets from remote storage
- Cache hits and misses
- Repeat bucket downloads

For more information on troubleshooting SmartStore, see Troubleshoot SmartStore in Managing Indexers and Clusters of Indexers.

Indexing: Indexer Clustering: Status

This topic is a reference for the `Indexer clustering: Status` dashboard in the Monitoring Console. See "About the Monitoring Console."

What does this view show?

This view is similar to the master dashboard on an indexer cluster master. See "View the master dashboard" in the Managing Indexers and Clusters of Indexers Manual.

Troubleshoot this view

You should be able to see data from all of your indexer clusters on this dashboard. If you have multiple indexer clusters and cannot see data from all of your indexer clusters, check that you have followed the setup steps for the Monitoring Console on a Splunk Enterprise deployment. Specifically,

- You are hosting the Monitoring Console on a search head that is a member of all of your clusters.
- You have labeled your indexer clusters.
- Each cluster master has been added as a search peer to the Monitoring Console.

Indexing: Indexer Clustering: Service Activity

This topic is a reference for the `Indexer clustering: Service activity` dashboard in the Monitoring Console. See "About the Monitoring Console."
What does this view show?

Several panels about indexer clustering.

Interpret results in this view

The panels in this dashboard might show no data. They show data only when bucket repair activity has occurred.

These panels help identify how many fixup tasks Splunk Enterprise has to undertake (that is, its backlog). A fixup task can result in more than one job.

The last two panels measure the time spent hitting the clustering endpoints.

Splunk Support and/or engineering might look at this data to profile the type of activity the cluster master is doing.

What to look for in this view

Use this view when an unforeseen event occurs in the cluster, such as a peer going down.

Ideally you will never have tasks pending.

If the search factor is not met, your search results will be incomplete.

If generation is not met, the entire cluster is unsearchable. This is unlikely to happen with clusters after 6.1.

After an event, you can watch this dashboard to make sure that the numbers of tasks (and jobs) trend down. For a healthy cluster, you want these panels to have no data.

Troubleshoot this view

Make sure you have completed all of the Monitoring Console setup steps.

In particular:

- Forward logs from search heads and deployers to indexers. See "Monitoring Console prerequisites."
- All indexer clustering dashboards need members of an indexer cluster. See "Set cluster labels."

Search: Search Activity

This topic is a reference for the Search Activity: Instance and Search Activity: Deployment dashboards in the Monitoring Console. See About the Monitoring Console.

What do these dashboards show?

The search activity dashboards provide an overview of search activity across your Splunk Enterprise deployment as well as more detailed information broken down by instance.
**Interpret results in these dashboards**

In the **Median Resource Usage of Searches** panels, note that:

- Resource usage is aggregated over all searches.
- Memory usage represents physical memory.
- In this chart, CPU usage is expressed in percentage of one core, not as system-wide CPU usage. As a result, you are likely to see values >100% here. This is not the case for other examples of CPU usage in the distributed management console.

In the **Aggregate Search Runtime** panel, note that:

- For each time bin in the chart, the Monitoring Console adds up the runtime of all searches that were running during that time range. Thus, you might see, for example, 1000 seconds of search in 5 minutes. This means that multiple searches were running over the course of those 5 minutes.
- For the modes historical batch and RT indexed, historical batch can be dispatched only by certain facilities within Splunk Enterprise (the scheduler, for example). RT indexed means indexed real-time.

In the **Top 10 Memory-Consuming Searches** panel, SID means search ID. If you are looking for information about a saved search, audit.log matches the name of your saved search (savedsearch_name) with its search ID (search_id), user, and time. With the search_id, you can look up that search elsewhere, like in the Splunk platform search logs (see What Splunk logs about itself).

The memory and CPU usage shown in these dashboards are for searches only. See the resource usage dashboards for all Splunk Enterprise resource usage.

In the **Instances by Median CPU Usage** panel, CPU can be greater than 100% because of multiple cores.

In the **Instances by Median Memory Usage** panel, memory is physical.

For the modes historical batch and RT indexed: historical batch can be dispatched only by certain facilities within Splunk Enterprise (the scheduler, for example). RT indexed means indexed real-time.

**What to look for in these dashboards**

Consider your search concurrency and resource usage compared to your system limits.

For information, see:

- Write better searches in the *Search Manual*.
- Get started with Search in the *Search Manual*.
- Configure the priority of scheduled reports in the *Reporting Manual*.
- Overview of summary-based search and pivot acceleration in the *Knowledge Manager Manual*.
- How concurrent users and searches impact performance in the *Capacity Planning Manual*.

A general pattern when looking over all the panels in this dashboards is to look for things that are close to exceeding their limits on your machines.
Troubleshoot these dashboards

The historical panels get data from introspection logs. If a panel is blank or missing information from non-indexers, check:

- that you are forwarding your introspection logs to your indexers, and
- the system requirements for platform instrumentation.

In the Search Activity: Instance > Search activity panel, the snapshots are taken every ten seconds by default. So if no searches are currently running, or if the searches you run are very short lived, the snapshots panel is blank and says "no results found."

Search: Search Usage Statistics

This topic is a reference for the Search Usage Statistics: Instance and Search Usage Statistics: Deployment dashboards in the Monitoring Console. See About the Monitoring Console.

What do these dashboards show?

These dashboards provide statistics about search usage across all search heads in your deployment as well as detail for each instance.

Interpret results in these dashboards

The following descriptions apply to the Long-Running Searches table in each dashboard:

- If "Earliest Time" or "Latest Time" is empty, it is probably a real-time search.
- If "Earliest Time" is "-", it means the earliest time is not specified and will be the start of computer epoch time. For most *nix environments, this is 1 January, 1970.
- If "Latest Time" is "-", it means the latest time is not specified and will be "now."

In Search Usage Statistics: Instance > Common Search Commands, runtimes are in seconds.

What to look for in these dashboards

A good practice is to look at long-running searches. You might find a search that you can optimize.

For more information, see Write better searches in the Search Manual.

Troubleshoot these dashboards

The historical panels in this view get their data from audit.log. If a panel is blank or missing information from non-indexers, verify that you are forwarding your introspection logs to your indexers.

The Long-Running Searches panel also uses information from a REST endpoint.
Search: KV Store

This topic is a reference for the KV Store: Deployment and KV Store: Instance dashboards in the Monitoring Console. See About the Monitoring Console.

What do these dashboards show?

The deployment-wide and instance-scoped KV store dashboards track much of the same statistics collected from KV store.

The KV store: Deployment dashboard in the Monitoring Console provides information aggregated across all KV stores in your Splunk Enterprise deployment. Instances are grouped by values of different metrics. For an instance to be included in this dashboard, it must be set with the server role of KV store. Do this in the Monitoring Console Setup page.

The instance level KV store view in the Monitoring Console shows performance information about a single Splunk Enterprise instance running the app key-value store. If you have configured the Monitoring Console in distributed mode, you can select which instance in your deployment to view.

Performance Metrics

Collection metrics come from the KVStoreCollectionStats component in the _introspection index, which is a historical record of the data at the /services/server/introspection/kvstore/collectionstats REST endpoint. The metrics are:

- Application. The application the collection belongs to.
- Collection. The name of the collection in KV store.
- Number of objects. The count of data objects stored in collection.
- Accelerations. The count of accelerations set up on the collection. Note: These are traditional database-style indexes used for performance and search acceleration.
- Accelerations size. The size in MBs of the indexes set up on the collection.
- Collection size. The size in MBs of all data stored in the collection.

Snapshots are collected through REST endpoints, which deliver the most recent information from the pertinent introspection components. The KV store instance snapshots use the endpoint /services/server/introspection/kvstore/serverstatus.

- Lock percentage. The percentage of KV store uptime that the system has held either global read or write locks. A high lock percentage has impacts across the board. It can starve replication or even make application calls slow, time out, or fail.
- Page fault percentage. The percentage of KV store operations that resulted in a page fault. A percentage close to 1 indicates poor system performance and is a leading indicator of continued sluggishness as KV store is forced to fallback on disk I/O rather than access data store efficiently in memory.
- Memory usage. The amount of resident, mapped, and virtual memory in use by KV store. Virtual memory usage is typically twice that of mapped memory for KV store. Virtual memory usage in excess of 3X mapped might indicate a memory leak.
- Network traffic. Total MBs in and out of KV store network traffic.
- Flush percentage. Percentage of a minute it takes KV store to flush all writes to disk. Closer to 1 indicates difficulty writing to disk or consistent large write operations. Some OSes can flush data faster than 60 seconds. In that case, this number can be small even if there is a writing bottleneck.
- Operations. Count of operations issued to KV store. Includes commands, updates, queries, deletes, getmores, and inserts. The introspection process issues a command to deliver KV store stats so the commands counter is typically higher than most other operations.
Current connections. Count of connections open on KV store.
Total queues. Total operations queued waiting for the lock.
Total asserts. Total number of asserts raised by KV store. A non-negative number can indicate a need to check KV store logs.

Historical

Many of the statistics in this section are present in the Snapshots section. The Historical view presents trend information for the metrics across a set span of time. These stats are collected in KVStoreServerStats. By default the Historical panels show information for the past 4 hours. Gaps in the historical graphs typically indicate a point at which KV store or Splunk Enterprise was unreachable.

- Memory usage - see above.
- Replication lag. The amount of time between the last operation recorded in the Primary OpLog and the last operation applied to a secondary node. Replication lag in excess of the primary opLog window could result in data not being properly replicated across all nodes of the replication set. In standalone instances without replication this panel does not return any results. Note: Replication lag is collected in the KVStoreReplicaSetStats component in the _introspection index.
- Operation count (average by minute) - see above. This panel shows individual operation types (for example, commands, updates, and deletes) or for all operations.
- Asserts - see above. This panel allows for filtering based on type of assert - message, regular, rollovers, user, warning.
- Lock percentage. Percentage of KV store uptime that the system has held global, read, or write locks. Filter this panel by type of lock held:
  - Read. Lock held for read operations.
  - Write. Lock held for write operations. KV store locking is "writer greedy," so write locks can make up the majority of the total locks on a collection.
  - Global. Lock held by the global system. KV store implements collection-level locks, reducing the need for aggressive use of the global lock.
- Page faults as a percentage of total operations - see above.
- Network traffic - see above. Added to this panel are requests made to the KV store.
- Queues over time. The number of queues, broken down by:
  - Read. Count of read operations waiting for a read lock to open.
  - Write. Count of write operations waiting for a write lock to open.
  - Total.
- Connections over time.
- Percent of each minute spent flushing to disk - see above.
- Slowest operations. The ten slowest operations logged by KV store in the selected time frame. If profiling is off for all collections, this could have no results even if you have very slow operations running. Enable profiling on a per collection basis in collections.conf.

Deployment Snapshots

Deployment Snapshot Statistics access the /services/server/introspection/kvstore/serverstatus REST endpoint.

Where do these dashboards get their data from?

KV store collects data in the _introspection index.

These statistics are broken into the following components:
**KVStoreServerStats.** Information about how the KV store process is performing as a whole. Polled every 27 seconds.

**KVStoreCollectionStats.** Information about collections within the KV store. Polled every 10 minutes.

**KVStoreReplicaSetStats.** Information about replication data across KV store Instances. Polled every 60 seconds.

**KVProfilingStats.** Information about slow operations. Polled every 5 seconds. Only available when profiling is enabled. **Note:** Enable profile only on development systems or for troubleshooting issues with KV store performance beyond what is available in the default panels. Profiling can negatively affect system performance and so should not be enabled in production environments.

In addition, KV store produces entries in a number of internal logs collected by Splunk Enterprise.

### Interpret these dashboards

<table>
<thead>
<tr>
<th>Panel</th>
<th>Critical</th>
<th>Warning</th>
<th>Normal</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Page faults per operation</td>
<td>1.3+</td>
<td>0.7–1.3</td>
<td>0–0.7</td>
<td>Measures how often read requests are not satisfied by what Splunk Enterprise has in memory, requiring Splunk Enterprise to contact the disk. Windows counts soft page faults, so Windows machines exhibit more page faults. Use lock percentage and queues instead.</td>
</tr>
<tr>
<td></td>
<td>Reads require heavy disk I/O, which could indicate a need for more RAM.</td>
<td>Reads regularly require disk I/O.</td>
<td>Reads rarely require disk I/O.</td>
<td></td>
</tr>
<tr>
<td>Lock percentage</td>
<td>50%+</td>
<td>30%–50%</td>
<td>0–30%</td>
<td>High lock percentage can starve replication and/or cause application calls to be slow, time out, or fail. High lock percentage typically means that heavy write activity is occurring on the node.</td>
</tr>
<tr>
<td>Network traffic</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>Network traffic should be commensurate with system use and application expectations. No default thresholds apply.</td>
</tr>
<tr>
<td>Replication latency</td>
<td>&gt;30 seconds</td>
<td>10–30 seconds</td>
<td>0–10 seconds</td>
<td>Replication needs are system dependent. Generally, replica set members should not fall significantly behind the KV captain. Replication latency over 30 seconds can indicate a mounting replication problem.</td>
</tr>
<tr>
<td>Primary operations log window</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
<td>Provided for reference. This is the amount of data, in terms of time, a system saved in the operations log for restoration.</td>
</tr>
<tr>
<td>Flushing rate</td>
<td>50%–100%</td>
<td>10%–50%</td>
<td>0–10%</td>
<td>A high flush rate indicates heavy write operations or sluggish system performance.</td>
</tr>
</tbody>
</table>

### Troubleshoot these dashboards

The historical panels get data from the _introspection and _internal indexes. Gaps in time in these panels indicate a time when KV store or Splunk Enterprise was unreachable. If a panel is completely blank or missing data from specific Splunk Enterprise instances, check:

- that you are forwarding your logs to your indexers, and
- the system requirements for platform instrumentation.

### Search: Scheduler Activity

This topic is a reference for the **Scheduler Activity** dashboards in the Monitoring Console. See About the Monitoring Console.
What do these dashboards show?

The **Scheduler activity: Deployment** dashboard shows an overview of the search or report scheduler. The instance dashboard shows more detailed information about the scheduler on a particular instance.

These dashboards show activity and success rate of the scheduler. That is, of all the searches that attempted to run, how many succeeded? A scheduled search can fail because of concurrency problems or because of a search workload.

The panels in these dashboards qualify the type of failure. Skip ratio and execution latency quantify the scheduler's performance.

The scheduler activity dashboards are useful whether or not you are using search head clustering. If you have a search head cluster, you can also use the **Search head clustering: Scheduler delegation** dashboard, which deals with how the captain orchestrates scheduler jobs.

Interpret results in these dashboards

In the deployment dashboard, the **Statistics** panel describes how the scheduler is performing per instance, but including all instances in the deployment. For example, maximum is the maximum on any individual instance in the deployment.

What to look for in these dashboards

If your scheduler reaches the maximum allowed concurrent searches, you will run into problems scheduling additional or long-running searches. See Configure the priority of scheduled reports for more information.

The snapshot quantities skip ratio and average execution latency should both be low.

The following is an example **Scheduler activity: Instance** panel for a scheduler that is skipping reports.
To find why the scheduler is skipping reports, scroll down to the panel labeled **Count of skipped reports by name and reason.**

**Troubleshoot these dashboards**

The scheduler activity dashboards require the monitored instances to be running Splunk Enterprise 6.3.0 or later.

Make sure you have completed all of the Monitoring Console setup steps.

**Search: Distributed Search**

This topic is a reference for the Monitoring Console dashboards related to distributed search. See About the Monitoring Console in this manual.

**What do these views show?**

The distributed search views expose the health, activity, and performance of the distributed search framework.

These views focus on communication between a search head and its peers during searches. In contrast, the search head clustering dashboards describe communication between search heads.

There are two basic ways to use these views:

1. Navigate to the health check at top of view, specific to this product area. Verify that these basic checks pass.

2. If your users report distributed search problems, use these views to understand how the components are performing. For example, users might see messages like "search peers could not participate in the search" or about search peers being unavailable or taking too long. For these types of messages, use these dashboards. If you know the instance reporting the problems, go directly to the Distributed search: Instance view. If not, start at the Distributed search: Deployment view. Look at the history of how these instances were behaving. These views can help you understand the distributed search framework. This hopefully gives you a better idea of the nature of the problem.

**Interpret results in these views**

For either view (Instance or Deployment), you can choose to examine search heads or search peers by selecting Search heads or Indexers at the top of the page. The metrics displayed on the dashboard change depending on which role you select.

On the Instance view, select a search head to see how the search head is communicating with its peers, from the operating context of this search head.

**What to look for in these views**

Scan for red flags in the Health Checks at the top of each view. The health checks are not comprehensive across the entire distributed search infrastructure. Rather, they are a high-level check of basic things.

The Snapshot panel exposes response times to a request and times for bundle replication. These times are vitals because they should take a very short time (under a second). Generally, if any of these times is a few seconds or longer, then something is not right.
In the **Deployment** view, select the search heads radial and use column sorting to inspect timing metrics:

- Dispatch directories are reaped per operation, so times over 15 seconds indicate problems.
- Bundle directory reaping should also be much less than 15 seconds.

The three **Heartbeat** metrics represent a vital on the search head. When they're high, the search peers might be oversubscribed and having trouble responding to communication requests in a timely manner. Response times over 1 second are not ideal and could indicate a developing problem. Response times over 5 or 10 seconds will start hitting up against timeouts. When this happens, searches actually fail. To continue troubleshooting, match this with the **Resource Usage: Machine** view corresponding to this peer. See Intermittent authentication timeouts on search peers in the **Troubleshooting Manual** for more information.

For additional help with distributed search problems, see General troubleshooting issues in the **Distributed Search Manual**.

### Troubleshoot these views

All of the metrics that these views leverage were introduced in Splunk Enterprise 6.3.0. If a component of your deployment is on a Splunk Enterprise version older than 6.3.0, these views will not include data from that component.

The snapshot panels use data from a variety of endpoints.

All historical panels in these views get their data from metrics.log.

### Search: Search Head Clustering

This topic is a reference for all of the Monitoring Console dashboards related to search head clustering. See About the Monitoring Console.

#### Status and Configuration

The **Status and Configuration** dashboard is an overview of your search head cluster. It is high-level information.

#### Configuration Replication

The **Configuration Replication** dashboard provides insight into configurations that a user changes on any search head cluster member (for example a new event type), and how these changes propagate through the cluster. Use this dashboard if you notice a significant lag in this propagation.

Action reference: The following are low-level actions exposed in the **Count of Actions Over Time** and **Time Spent on Actions Over Time** panels. These panels can be helpful for troubleshooting.

<table>
<thead>
<tr>
<th>Action</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>accept_push</td>
<td>On the captain, accept replicated changes from a member.</td>
</tr>
<tr>
<td>acquire_mutex</td>
<td>Acquire a mutex (mutual exclusion) that &quot;protects&quot; the configuration system.</td>
</tr>
<tr>
<td>add_commit</td>
<td>On a member, record a change.</td>
</tr>
<tr>
<td>base_initialize</td>
<td>Initialize a configuration &quot;root&quot; (e.g. $SPLUNK_HOME/etc).</td>
</tr>
<tr>
<td>Action</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>check_range</td>
<td>Compare two ranges of configuration changes.</td>
</tr>
<tr>
<td>compute_common</td>
<td>Find the latest common change between a member and the captain.</td>
</tr>
<tr>
<td>pull_from</td>
<td>On a member, pull changes from the captain.</td>
</tr>
<tr>
<td>purge_eligible</td>
<td>On a member, purge sufficiently old changes from the repo.</td>
</tr>
<tr>
<td>push_to</td>
<td>On a member, push changes to the captain.</td>
</tr>
<tr>
<td>release_and_reacquire_mutex</td>
<td>Release, then re-acquire a mutex that “protects” the configuration system. This is similar to acquire_mutex.</td>
</tr>
<tr>
<td>reply_pull</td>
<td>On the captain, reply to a member's pull_from request.</td>
</tr>
<tr>
<td>repo_initialize</td>
<td>Initialize a configuration repo (from disk).</td>
</tr>
</tbody>
</table>

We expect this information to be leveraged by Splunk Support. If you have issues with configuration replication, you can look at this dashboard for clues. But we expect you to use this dashboard more for gathering information after you file your Support case, rather than gaining insight on your own.

**Artifact Replication**

The **Artifact Replication** dashboard contains several panels describing the cluster’s backlog of search artifacts to replicate. See Search head clustering architecture in the 'Distributed Search' Manual.

The **Warnings and Errors Patterns** panel groups warning and error events based on text within the messages. The grouping functionality uses the cluster command.

If your search head cluster is replicating artifacts on time, its **Count of Artifacts Awaiting Replication** will be at or near zero. A few artifacts awaiting replication is likely not a warning sign. A consistently high and especially a growing number of artifacts could indicate a problem with replication. If you have many artifacts waiting, someone using another search head might not get a local cache and will experience slowness in search availability.

**Median Count of Artifacts to Replicate** is (as advertised) a median. This means that if you have narrow spikes, you won’t see them at larger time ranges.

The **Artifact Replication Job Activity** panel shows the rate of change of replicating jobs (specifically, the backlog change is the rate). The backlog change can be negative, if your cluster is catching up with its backlog. In this panel, a red flag to look for is a backlog that grows consistently (that is, if the backlog change is always positive). If this happens, the **Median Count of Artifacts to Replicate** panel above shows a continually growing backlog.

**Scheduler Delegation**

See Search head clustering architecture in the Distributed Search Manual.

In the **Scheduler Status** panel, note that max_pending and max_running are “highwater marks” over a 30 second period. That is, they are the highest number of jobs that were pending or running in a 30 second span. You can select one of several functions in this panel. The "maximum" function works in a straightforward manner with these statistics. But take a moment to think through what "average," "median," or "90th percentile" mean. For example: Say max_pending is 4 over 30 seconds, then you average the values of max_pending. You end up with the average high values, not the average of all. So if the number of pending jobs fluctuates a lot, the average max_pending might not be close to a straight average of the number of pending jobs.
**App Deployment**

The App Deployment dashboard monitors apps as they are deployed from a deployer to search head cluster members.

See About deployment server and forwarder management in the *Updating Splunk Enterprise Instances Manual*.

In the **Apps status** panel, a persistent discrepancy indicates that the deployer has not finished deploying apps to its members.

**Troubleshoot these dashboards**

The search head clustering dashboards require the monitored instances to be running Splunk Enterprise 6.2.0 or greater.

Make sure you have completed all of the Monitoring Console setup steps.

In particular:

- Forward logs from search heads and deployers to indexers. See Monitoring Console prerequisites.
- For all search head clustering dashboards, search heads need to be set as search peers of the Monitoring Console.
- All search head clustering dashboards need members of a search head cluster. See Set cluster labels. Note that the app deployer also needs a label.

For the App Deployment dashboard:

- The deployer needs to be a search peer of the Monitoring Console, or the Monitoring Console can be hosted on the deployer. See Add instances as search peers.
- The deployer needs to have the deployer role (it might auto-detect). Check this in Monitoring Console > Settings > General Setup.
- The deployer needs to be manually labeled as a member of the SHC. (It will not auto-detect.) Set this in Monitoring Console > Settings > General Setup.
- The deployer must forward logs, as above. See Monitoring Console prerequisites.

**Resource Usage**

This topic is a reference for the Resource Usage dashboards in the Monitoring Console. See About the Monitoring Console.

**What do these dashboards show?**

There are several Resource Usage dashboards, which you access through the **Resource Usage** menu. The Resource Usage: Deployment dashboard provides deployment-wide resource information, such as CPU usage, physical memory usage, and disk usage. These panels can be useful for capacity planning.

The other dashboards provide usage information by instance or machine.

**Interpret results in these dashboards**

About physical memory usage in these dashboards: on Linux, the OS uses free physical memory to cache filesystem
resources. But memory for this is loosely bound, and the OS frees it up if a higher priority process needs it. The Monitoring Console reporting cannot discern how much memory is loosely locked up in this way.

The historical data in these dashboards comes from `resource_usage.log` in the `.introspection` index. See What data gets logged in the platform instrumentation chapter of the Troubleshooting Manual.

**Resource usage: Deployment**

The Deployment-Wide Median Disk Usage panel takes into account all partitions in use by each Splunk Enterprise instance.

**Resource usage: Machine**

This dashboard can be useful for operational post mortems, as well as for capacity planning. See the Capacity Planning Manual for more information.

In the Median CPU Usage panel, 100% means the entire system, however many cores the system has. This is in contrast to the Search Activity dashboards, where 100% means one core.

The disk space in this dashboard refers only to partitions with a Splunk Enterprise instance on them.

**Resource usage: Instance**

In the two "process class" panels, the value of process class can be splunkd server, search, Splunk Web, index service, scripted input, KVStore, or other.

A process class is an aggregate of processes within one class. For more information about

- splunkd, see Splunk Enterprise architecture and processes in the Installation Manual.
- search, see Get started with Search and Write better searches in the Search Manual.
- splunkweb, see Splunk Enterprise architecture and processes in the Installation Manual.
- scripted input, see Get data from APIs and other remote data interfaces through scripted inputs in the Getting Data In Manual.
- KVStore, see the KV store dashboards dashboard in the Monitoring Console.

The index service consists of housekeeping tasks related to indexing. These tasks run at the end of the indexing pipeline but are asynchronous. These processes run on their own, not through splunkd.

The Disk Usage and Median Disk Usage panels list only the partitions that Splunk Enterprise uses.

**What to look out for in these dashboards**

The panels of the deployment-wide dashboard group instances by value ranges. In this dashboard, look for outliers: instances that are not like the others.

In all of the resource usage dashboards, look for patterns that appear over time. For example, in the instance and machine dashboards, look for memory usage increasing over time without recovering.

In the Instance dashboard, if a process class that is using a lot of resources turns out to be search, investigate further by going to the Search activity: Instance dashboard.
Troubleshoot these dashboards

The historical panels get data from introspection logs. If a panel is blank or missing information from non-indexers, check:

- that you are forwarding your introspection logs to your indexers, and
- the system requirements for platform instrumentation.

Forwarders

This topic is a reference for the Forwarders: Deployment, Forwarders: Instance, and the Splunk TCP Input Performance deployment and instance dashboards in the Monitoring Console. See About the Monitoring Console in this manual.

What do these views show?

The Monitoring Console monitors forwarder connections (in the Forwarders dashboards) and communication (in the Splunk TCP Input dashboards).

The Splunk TCP Input views monitor Splunk TCP Inputs, that is, data from one Splunk instance to another. Usually this is a forwarder sending data to an indexer. These views do not monitor TCP input from a non-Splunk device to a collector, like an Apache server sending its logs to a forwarder.

Interpret results in these views

Forwarders: Deployment view

The Status panel can show the value "active" or "missing". When the scheduled search runs to update this panel, it looks back 15 minutes. If a forwarder connects to the indexers in those 15 minutes, then its status is "active." If not, its status is "missing." To permanently remove missing forwarders from your dashboards, rebuild the forwarder asset table. See Configure forwarder monitoring in this manual.

This lookback time is different from the data collection interval (in Settings > Forwarder Monitoring Setup), which is how often that scheduled search runs. Read about time settings in Configure forwarder monitoring in this manual.

In the Status and configuration panel, the time shown is the last time that the scheduled search completed.

Forwarders: Instance view

The quantity called "outgoing data rate" measure the data received by an indexer from a forwarder. This measurement comes from metrics.log on the indexer. See About metrics.log in the Troubleshooting Manual.

If you can't find your indexed data in Splunk Enterprise, you can look at Monitoring Console dashboards in this order:

1. Forwarder views.
2. Splunk TCP input views.
3. Indexing views.

See Troubleshoot forwarder/receiver connection in the Forwarding Data manual.
What to look for in these views

Start at the **Forwarders: Deployment** view to see whether your forwarders are reporting as expected, or whether one of them is missing.

This dashboard is paired with a preconfigured platform alert, which can notify you when one or more forwarders is missing.

Troubleshoot these views

**Forwarders and Splunk TCP Input dashboards**

If these dashboards lack data, verify that you have completed all of the setup steps for the Monitoring Console, in either distributed or standalone mode.

Like all Monitoring Console dashboards, these dashboards need metrics.log from the indexers. The Monitoring Console does not query forwarders directly for data, but rather gets its data from the indexers that the forwarders connect to.

**Steps specific to the Forwarders dashboards**

For any of the **Forwarders: Deployment** or **Forwarders: Instance** dashboard panels to work, you must follow the setup steps in Configure forwarder monitoring in this manual. Note the prerequisite that the historical panels need forwarders with individual GUIDs.

Averages on the **Forwarders** dashboards are not calculated until at least one of the "data collection intervals" (as defined in Monitoring Console > Settings > Forwarder monitoring setup) elapses.
Proactive Splunk component monitoring

About proactive Splunk component monitoring

Proactive Splunk component monitoring lets you view the health of Splunk Enterprise features from the output of a REST API endpoint. Individual features report their health status through a tree structure that provides a continuous, real-time view of the health of your deployment.

You can access feature health information using the splunkd health report in Splunk Web. See View the splunkd health report.

You can also access feature health information programmatically from the server/health/splunkd endpoint. See Query the server/health/splunkd endpoint.

For more information on the splunkd process, see Splunk Enterprise Processes.

How it works

Proactive Splunk component monitoring records the health status of splunkd in a tree structure, where leaf nodes represent particular Splunk Enterprise features, and intermediary nodes categorize the various features. Feature health status is color-coded in three states:

- Green: The feature is functioning properly.
- Yellow: The feature is experiencing a problem.
- Red: The feature has a severe issue and is negatively impacting the functionality of your deployment.
### Health Status of Splunkd

<table>
<thead>
<tr>
<th>Health status tree node</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>splunkd</td>
<td>The top level node of the status tree shows the overall health status (color) of splunkd. The status of splunkd shows the least healthy state present in the tree. The REST endpoint retrieves the instance health from the splunkd node.</td>
</tr>
<tr>
<td>Feature categories</td>
<td>Feature categories represent the second level in the health status tree. Feature categories are logical groupings of features. For example, &quot;BatchReader&quot; and &quot;TailReader&quot; are features that form a logical grouping with the name &quot;File Monitor Input&quot;. Feature categories act as buckets for groups of features, and do not have their own health status.</td>
</tr>
<tr>
<td>Features</td>
<td>The next level in the status tree is feature nodes. Each node contains information on the health status of a particular feature. Each feature contains one or more indicators that determine the status of the feature. The overall health status of a feature is based on the least healthy color of any of its indicators.</td>
</tr>
<tr>
<td>Indicators</td>
<td>Indicators are the fundamental elements of the splunkd health report. These are the lowest levels of functionality that are tracked by each feature, and change colors as functionality changes. Indicator values are measured against red or yellow threshold values to determine the status of the feature. See What determines the status of a feature?</td>
</tr>
</tbody>
</table>

For a list of supported Splunk Enterprise features, see Supported features.

**What determines the status of a feature?**

The current status of a feature in the status tree depends on the value of its associated indicators. Indicators have configurable thresholds for yellow and red. When an indicator's value meets threshold conditions, the feature's status changes.
For information on how to configure indicator thresholds, see Set feature indicator thresholds.

For information on how to troubleshoot the root cause of feature status changes, see Investigate feature health status changes.

**Default health status alerts**

By default, each feature in the splunkd health report generates an alert when a status change occurs, for example from green to yellow, or yellow to red. You can enable/disable alerts for any feature and set up alert notifications via email, PagerDuty, or web hook in health.conf or via REST endpoint. For more information, see Configure health report alerts.

**Health status viewpoint**

The splunkd status tree shows the health of your Splunk Enterprise deployment from the viewpoint of the instance type that you are monitoring. For example, in an indexer cluster environment, the cluster master and peer nodes each show a different set of features contributing to the overall health of splunkd.

**Requirements**

Proactive Splunk component monitoring has the following requirements and limitations.

**Splunk Enterprise version requirements**

Proactive Splunk component monitoring requires Splunk Enterprise version 7.1 or higher.

**Operating system requirements**

Proactive Splunk component monitoring is available on all operating systems supported by Splunk Enterprise. For a list of supported operating systems, see System requirements in the Installation Manual.

**REST endpoint access requirements**

Proactive Splunk component monitoring displays information from the server/health/splunkd endpoint. You can view this information in the splunkd health report in Splunk Web.

To query the server/health/splunkd endpoint directly requires access to the splunkd management port (default port 8089) over http. For more information, see Connecting to splunkd in the REST API User Manual.

For endpoint details, see server/health/splunkd in the REST API Reference Manual.

**Configure the splunkd health report**

The splunkd health report displays the status of a pre-defined set of Splunk Enterprise features. You can modify some health report settings, including feature thresholds, using the health report manager page in Splunk Web, or by editing health.conf.

For more information on health report configuration settings in health.conf, see health.conf.spec in the Admin Manual.
Supported features

The splunkd health report lets you monitor these Splunk Enterprise features:

<table>
<thead>
<tr>
<th>Feature Category</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Forwarding / Splunk-2-Splunk Forwarding</td>
<td>TCPOutAutoLB</td>
</tr>
<tr>
<td>File Monitor Input</td>
<td>BatchReader, TailReader</td>
</tr>
<tr>
<td>Index Processor</td>
<td>Buckets, Disk Space, Index Optimization</td>
</tr>
<tr>
<td>Indexer Clustering</td>
<td>Cluster Bundles, Data Durability, Data Searchable, Indexers, Indexing Ready, Master Connectivity, Replication Failures, Slave State, Slave Version, Search Head Connectivity</td>
</tr>
<tr>
<td>Search Head Clustering</td>
<td>Member to Captain Connection, Captain Common Baseline, Captain Election Overview, Members Overview, Snapshot Creation</td>
</tr>
</tbody>
</table>

For more information on the above features, see $SPLUNK_HOME/etc/system/default/health.conf.

Set feature indicator thresholds

Each feature in the health status tree has one or more indicators. Each indicator reports a value against a pre-set threshold, which determines the status of the feature. When the indicator value meets the threshold condition, the health status of the feature changes, for example, from green to yellow, or yellow to red.

There are two valid thresholds for each indicator: **yellow** and **red**. You can modify threshold values for any feature using Splunk Web or health.conf.

Set thresholds using Splunk Web

To set feature threshold values in Splunk Web:

1. Log in to Splunk Web on the instance you are monitoring.
2. Click Settings > Health report manager.
3. For the feature you want to modify, click Edit Thresholds.
4. Set new indicator threshold values. For example, to modify thresholds for the Batch Reader feature, set new Red or Yellow threshold values for the `data_out_rate` indicator:
5. Click **Save**.

To view and edit threshold settings on the health report manager page, your role must be assigned `list_health` and `edit_health` capabilities. For more information, see Set access controls for the splunkd health report.

**Set thresholds using health.conf**

To set feature threshold values in `health.conf`:

1. Log in to the instance you are monitoring.
2. Edit `$SPLUNK_HOME/etc/system/local/health.conf`
3. In the feature stanza, set new indicator threshold values. For example, to modify indicator threshold values for the `batchreader` feature, set new values for `data_out_rate:yellow` and `data_out_rate:red` thresholds in the following stanza:

   ```
   [feature:batchreader]
   indicator:data_out_rate:red = 10
   indicator:data_out_rate:yellow = 5
   ```

   Indicator thresholds are pre-set to values that apply to most use cases. When you modify threshold values, make changes in small increments. Setting threshold values too high can mask serious problems or failures.

   For detailed descriptions of each feature indicator, see `$SPLUNK_HOME/etc/system/default/health.conf`.

**Disable a feature**

You can disable any feature in `health.conf`. Disabling a feature removes that feature from the splunkd health status tree. This is useful, for example, if you want to exclude a feature’s status from the health report, while you troubleshoot a problem with that feature. All supported features are enabled by default in `health.conf`.

There are three ways to disable a feature:

- Disable the feature in Splunk Web.
- Edit the feature stanza in `health.conf`.
- Use the `/server/health-config` endpoint.

**Disable a feature in Splunk Web**

1. Log in to Splunk Web on the instance you are monitoring.
2. Click **Settings > Health report manager**.
3. Set the switch to disable for the particular feature.
   
   The feature is disabled and no longer impacts the overall health status of splunkd.

**Disable a feature in health.conf**

1. Log in to the instance you are monitoring.
2. Edit `$SPLUNK_HOME/etc/system/local/health.conf`
3. In the feature stanza, add `disabled = 1`. For example, to disable the Data Durability feature:

   ```
   [feature:data_durability]
   indicator:cluster_replication_factor:red = 1
   indicator:cluster_search_factor:red = 1
   disabled = 1
   ```
To enable a feature, set `disabled = 0`

4. **Reload** `health.conf`:

    curl -k -u admin:pass https://<host>:<mPort>/services/configs/conf-health.conf/_reload

---

**Disable a feature using REST endpoint**

1. Log in to the instance you are monitoring.
2. Run the following command against the `server/health-config/{feature_name}` endpoint. For example, to disable the `batchreader` feature:

    curl -k -u admin:pass \
    https://<host>:<mPort>/services/server/health-config/feature:batchreader -d disabled=1
3. Validate the feature no longer appears in the `splunkd` status report in Splunk Web.

For endpoint details, see `server/health-config/{feature_name}` in the **REST API Reference Manual**.

To access `server/health-config/` endpoints, your role must have the `edit_health` capability.

---

**Suppress health status updates**

Features in the health status tree update their status at predetermined intervals. A feature whose health status changes frequently can cause excessive undesirable changes to the overall status of the `splunkd` health report. To prevent this, use the `suppress_status_update_ms` attribute in `health.conf` to reduce the frequency with which a particular feature can update its health status.

Use the `suppress_health_status_update_ms` attribute to:

- Limit excessive changes to the internal state by individual features.
- Reduce the number of log entries that arise from rapid feature status changes.
- Help quiet “noisy” features.

For example, an indexer clustering feature, such as `data_durability`, can experience frequent status changes during operations that impact its indicators: `cluster_replication_factor` and `cluster_search_factor`. To avoid frequent changes to the overall `splunkd` health report, you might set `suppress_status_update_ms = 60000` to reduce health status updates to once every minute.

To suppress health status updates:

1. Log in to the instance you are monitoring.
2. **Edit** `$SPLUNK_HOME/etc/system/local/health.conf`
3. In the appropriate feature stanza, add the `suppress_status_update_ms` attribute. For example:

    ```
    [feature:data_durability]
    indicator:cluster_replication_factor:red = 1
    indicator:cluster_search_factor:red = 1
    suppress_status_update_ms = 60000
    ```

    By default, the minimum amount of time that must elapse between status updates is 300ms.

For more information, see `health.conf.spec` in the **Admin Manual**.
Configure health status logs

Each feature in the splunkd health status tree generates log entries in health.log. These log entries record information about feature indicator status changes over time. health.log is located in SPLUNK_HOME/var/log/splunk/.

There are two types of health.log log entries:

HealthChangeReporter: This log entry records specific health status changes for a feature indicator. Each entry includes a timestamp, feature name, indicator name, previous color, new color, and a possible reason for the status change. This log entry appears only if a feature's status changes, for example, from green to red:

```
02-28-2018 20:26:52.775 +0000 INFO  HealthChangeReporter - feature="Data Durability" indicator="cluster_replication_factor" previous_color=green color=red reason="Replication Factor is not met"
```

PeriodicHealthReporter: This log entry keeps an ongoing record of the status of each feature in the health status tree. Each entry includes a timestamp, the feature name, and current color. Log entries are made at a user-configurable interval. For example:

```
02-28-2018 20:27:06.826 +0000 INFO  PeriodicHealthReporter - feature="Data Durability" color=red
```

Set health.log entry intervals

You can set the interval at which PeriodicHealthReporter log entries are added to health.log. This is useful if you want to increase or decrease the overall number of log entries that appear in health.log.

To adjust the frequency of PeriodicHealthReporter log entries in health.log:

1. Log in to the instance you are monitoring.
2. Edit $SPLUNK_HOME/etc/system/local/health.conf
3. In the [health_reporter] stanza, set the full_health_log_interval attribute to an appropriate value in seconds. For example:
   ```
   [health_reporter]
   full_health_log_interval = 60
   ```
   By default, each feature generates a PeriodicHealthReporter log entry every 30 seconds.

Set up alerts for the splunkd health report

The splunkd health report generates alerts for all features in the health status tree. When a feature indicator meets the threshold condition, the feature's health status changes, for example from green to red, and an alert fires. Use health status alerts to maintain visibility into the health of your deployment, whether or not you are logged into Splunk Web.

You can configure health report alerts as follows:

- Enable/disable alerts on the global, feature, or indicator level.
- Send alert notifications via email or PagerDuty.
- Set the health status color (yellow or red) that triggers an alert.
- Set a minimum duration that must elapse between alerts.

You can configure health report alerts by directly editing health.conf or querying the server/health-config endpoint.
Disable health report alerts

Alerts are enabled by default for all features in the splunkd health report. You can disable alerts at the global, feature, or indicator level. Disabling alerts at the global level overrides enabled alerts at the feature level. Likewise, disabling alerts at the feature level overrides enabled alerts at the indicator level.

Disabling alerts is useful for reducing noise from non-critical features and minimizing false positives when performing maintenance tasks.

Disable alerts using health.conf

To disable alerts for all features in the splunkd health report:

1. Edit $SPLUNK_HOME/etc/system/local/health.conf.
2. In the health_reporter stanza, set alert.disabled = 1. For example:

   ```
   [health_reporter]
   full_health_log_interveral = 30
   suppress_status_update = 600
   alert.disabled = 1
   
   To enable alerts for all features in the splunkd health report, set alert.disabled = 0.
   ```

To disable alerts for a single feature:

1. Edit $SPLUNK_HOME/etc/system/local/health.conf.
2. In the stanza for the particular feature, set alert.disabled = 1. For example:

   ```
   [feature:indexers]
   ...
   indicator:missing_peers:yellow = 1
   indicator:missing_peers:red = 1
   alert.disabled = 1
   
   To enable alerts for a feature, set alert.disabled = 0.
   ```

To disable alerts for a single feature indicator:

1. Edit $SPLUNK_HOME/etc/system/local/health.conf.
2. In the stanza for a particular feature, set alert:<indicator_name>.disabled = 1. For example, in the following stanza, alerting for the indicator s2s_connections is disabled:

   ```
   [feature:s2s_autolb]
   ...
   indicator:s2s_connections:yellow = 20
   indicator:s2s_connections:red = 70
   alert:s2s_connections.disabled = 1
   
   To enable alerts for an indicator, set alert:.disabled = 0.
   ```

Disable alerts using REST endpoint

To disable alerts for features and indicators, send a POST request to server/health-config/{feature_name}. For example, to disable alerts for the batchreader feature on the instance you are monitoring run the following command:

```bash
curl -k -u admin:pass https://<host>:<mPort>/services/server/health-config/feature:batchreader -d alert.disabled=1
```
For endpoint details, see server/health-config/{feature_name} in the REST API Reference Manual.

To access server/health-config endpoints, a role must have the edit_health capability.

**Set up health report alert actions**

You can set up alert actions that run when an alert fires, such as sending alert notifications via email, mobile device, or PagerDuty.

Alert actions apply on the global level only. Multiple alert actions for the same action type are not supported. For example, you cannot have multiple email actions and multiple PagerDuty actions.

Before you can send health email alert notifications, you must configure email notification settings in Splunk Web. For instructions, see Email notification action in the "Alerting Manual".

**Set up email notifications in health.conf**

To set up email alert notifications:

1. Edit SPLUNK_HOME/etc/system/local/health.conf
2. Add the [alert_action:email] stanza and specify the recipients. For example:

   ```
   [alert_action:email]
   disabled = 0
   action.to = <recipient@example.com>
   action.cc = <recipient_2@example.com>
   action.bcc = <other_recipients@example.com>
   ```

**Set up PagerDuty notifications in health.conf**

Before you can send alert notifications to PagerDuty, you must install the PagerDuty App from Splunkbase. You must also add a new service to your PagerDuty integration, and copy the integration key. For more information, see PagerDuty documentation.

To set up PagerDuty alert notifications:

1. Edit $SPLUNK_HOME/etc/system/local/health.conf.
2. Add the [alert_action:pagerduty] stanza and specify the integration key. For example:

   ```
   [alert_action:pagerduty]
   disabled = 0
   action.integration_url_override = <integration key>
   ```

For more information, see health.conf.example.

**Set up alert notifications using REST**

To set up alert notifications, send a POST request to server/health-config/{alert_action}. For example, to set up an email alert notification:

```bash
curl -k -u admin:pass https://localhost:8089/services/server/health-config/alert_action:email -d action.to=admin@example.com -d action.cc=admin2@example.com
```

For endpoint details, see server/health-config/{alert_action} in the REST API Reference Manual.
Set the alert threshold color

You can set the threshold color that triggers an alert. Possible alert threshold values are yellow or red. If the threshold value is yellow, an alert fires for both yellow and red. If the value is red, an alert fires for red only. The default alert threshold value is red.

Set the alert threshold color in health.conf

To set the alert threshold color on the global or feature level:

1. Edit `$SPLUNK_HOME/etc/system/local/health.conf`.
2. Add the `alert.threshold_color` setting to the `[health_reporter]` or `[feature:<feature_name>]` stanza. For example:

```bash
[feature:replication_failures]
...  
alert.threshold_color = yellow
indicator:replication_failures:red = 10
indicator:replication_failures:yellow = 5
```

To set the alert threshold color on the indicator level:

1. Edit `$SPLUNK_HOME/etc/system/local/health.conf`.
2. Add the `alert:<indicator name>.threshold_color` setting to the feature stanza. For example:

```bash
[feature:replication_failures]
...  
indicator:replication_failures:red = 10
indicator:replication_failures:yellow = 5
alert:replication_failures.threshold_color = yellow
```

Alert threshold color settings at the indicator level override alert threshold color settings at the feature level.

Set the alert threshold color using REST

To set the alert threshold color for a feature or indicator, send a POST request to `server/health-config{feature_name}`. For example, to set the alert threshold color for the Replication Failures feature:

```bash
```

For endpoint details, see `server/health-config/{feature_name}` in the REST API Reference Manual.

Set minimum duration between alerts

You can set the amount of time an unhealthy health status persists before an alert fires using the `alert.min_duration_sec` setting. You can use this setting to help reduce noise from feature health status changes that might be rapidly flipping between states, for example, between green and yellow or yellow and red.

Set minimum duration between alerts in health.conf

To set the minimum duration between alerts on the global or feature level:

1. Edit `$SPLUNK_HOME/etc/system/local/health.conf`.

48
2. Add the `alert.min_duration_sec` setting to the `[health_reporter]` or `[feature:<feature_name>]` stanza. For example:

```
[feature:replication_failures]
...
alert.min_duration_sec = 600
indicator:replication_failures:red = 10
indicator:replication_failures:yellow = 5
```

To set the minimum duration between alerts on the indicator level:

1. Edit `$SPLUNK_HOME/etc/system/local/health.conf`.
2. Add the `alert:<indicator name>.min_duration_sec` setting to the `[feature:<feature_name>]` stanza. For example:

```
[feature:replication_failures]
...
indicator:replication_failures:red = 10
indicator:replication_failures:yellow = 5
alert:replication_factor.min_duration_sec = 600
```

Minimum duration between alerts settings on the feature level override settings on the indicator level.

**Set minimum duration between alerts using REST**

To set the minimum duration between alerts for a feature or indicator, send a POST request to `server/health-config/{feature_name}`. For example, to set the minimum duration between alerts for the Replication Failures feature:

```
curl -k -u admin:pass https://localhost:8089/services/server/health-config/feature:replication_failures -d
alert.min_duration_sec=600
```

For endpoint details, see `server/health-config/{feature_name}` in the REST API Reference Manual.

**Set access controls for the splunkd health report**

To monitor, edit, and enable splunkd health report features, the user's role must be assigned the appropriate capabilities. The following health report capabilities are enabled by default for `role_admin`:

<table>
<thead>
<tr>
<th>Capability</th>
<th>Permissions granted to role</th>
</tr>
</thead>
<tbody>
<tr>
<td>list_health</td>
<td>View the splunkd health report and health manager page in Splunk Web.</td>
</tr>
<tr>
<td>edit_health</td>
<td>Edit and enable health report features in Splunk Web or through REST endpoint.</td>
</tr>
</tbody>
</table>

You can add the above capabilities to any role in Splunk Web or in `authorize.conf`.

To add a capability to a role in Splunk Web, see Add and edit roles with Splunk Web.

To add a capability to a role in `authorize.conf`, see Add and edit roles with `authorize.conf`. 
Investigate feature health status changes

There are two ways to access feature health status information from the `/server/health/splunkd` endpoint:

- View the `splunkd` health report in Splunk Web.
- Query the `server/health/splunkd` endpoint.

You can also monitor feature health status changes in `$SPLUNK_HOME/var/log/health.log`. For more information on `health.log` file entries, see Configure health status logs.

View the `splunkd` health report

The `splunkd` health report lets you view the current status of features in the `splunkd` health status tree. You can use the report to identify features whose status indicates a problem, and investigate the cause of those problems.

1. On the instance you want to monitor, log in to Splunk Web.
2. In the main menu, check the color of the health report icon. The color of this icon indicates the overall status of `splunkd`.
3. Click the health report icon to open the health report.
4. In the health status tree, click on any feature to view information about the feature's status.
5. For features in the red or yellow state, review Root Cause and Last 50 related messages for information that can help you identify the cause of a feature's status change.

Example

This example shows how you can use the splunkd health report to investigate feature health status changes on a cluster master instance.

1. Review the splunkd health report.
2. Investigate root cause and related messages.
3. Confirm the cause of feature status change.

1. Review the splunkd health report

1. On the cluster master instance, log in to Splunk Web.
2. Check the color of the health report icon.

The red icon indicates that one or more features in the splunkd health status tree has a severe issue.
3. Click the health report icon to open the health report.
The health report shows the data_durability feature has a severe issue.

2. Investigate root cause and related messages

In the health status tree, click the data_durability feature. The following diagnostic information appears:

**Root Cause:** "Replication Factor is not met."

If the number of peers in the cluster is less than the replication factor, the replication factor cannot be met. Therefore a possible cause of the feature's red status is an offline peer.

**Last 50 Related Messages:** Searching the related messages, you see log entries that contain streaming errors. For example:

```
The streaming error suggests that bucket replication is failing because a source peer cannot communicate with a target peer. This type of error can be caused by a network interruption or an offline peer.
```

3. Confirm the cause of feature status change

After you use the splunkd health report to investigate the cause of the feature's status change, which suggests a peer is offline, you can use the Monitoring Console to check the status of cluster peers and confirm if the suspected cause is correct.

1. In Splunk Web, click **Settings > Monitoring Console**.
2. Click **Indexing > Indexer Clustering > Indexer Clustering: Status**.

   The dashboard shows that the peer node **idx3** is stopped.

   ![Dashboard showing stopped idx3 peer](image)

3. Click on the **idx3** peer to see more information. The **idx3** peer GUID matches the GUID of the target peer in the streaming error message. This confirms that the cause of the data_durability feature's red status is an offline peer.

   ![Idx3 peer details](image)

52
You can now take steps to restart the peer and have it rejoin the cluster, which returns the data_durability feature to the green state.

**Query the server/health/splunkd endpoint**

You can integrate proactive Splunk component monitoring with existing third-party monitoring tools and other applications.

To see the overall health status of splunkd, query the server/health/splunkd endpoint. For example:

```bash
curl -k -u admin:pass https://<host>:8089/services/server/health/splunkd
```

For endpoint details, see server/health/splunkd in the *REST API Reference Manual*.

To see the overall health of splunkd, as well as the health of each feature reporting to the status tree, and information on feature health status changes, query server/health/splunkd/details. For example:

```bash
curl -k -u admin:pass https://<host>:8089/services/server/health/splunkd/details
```

For endpoint details, see server/health/splunkd/details in the *REST API Reference Manual*.

For more information on how to use Splunk REST API endpoints, see Connecting to splunkd in the *REST API User Manual*.

For information on Splunk SDKs that support the Splunk REST API, see Overview of Splunk SDKs.