Splunk® Enterprise 8.2.0

管理索引器和索引器群集

生成时间：2021 年 5 月 24 日，14:30

Table of Contents

索引概述

索引、索引器和索引器群集

本手册介绍 Splunk Enterprise 数据存储库以及创建和管理它们的 Splunk Enterprise 组件。

索引是 Splunk Enterprise 数据的存储库。Splunk Enterprise 将传入数据转换为**事件**，然后将其存储在索引中。

索引器是用于为数据创建索引的 Splunk Enterprise 实例。对于较小部署，单个实例可能还会执行其他 Splunk Enterprise 功能，如数据导入和搜索管理。但在大型、分布式**部署**中，数据导入和搜索管理功能分配给其他 Splunk Enterprise 组件。本手册专门介绍在单个实例或分布式部署上下文中的索引功能。

索引器群集是配置为复制彼此数据的一组索引器，这样系统便会保留所有数据的多个副本。此过程称为**索引复制**或索引器群集化。通过保留数据的多个相同副本，群集能够阻止数据丢失，同时还便于数据搜索。

索引

当 Splunk Enterprise 处理传入数据时，会将数据添加到索引。Splunk Enterprise 随附了几个索引，您您可以根据自身需要创建其他索引。

Splunk Enterprise 索引包含各种文件。这些文件可分为两种主要类别：

- 压缩形式的原始数据（**原始数据**）
- 指向原始数据的索引（**索引文件**，也称为 **tsidx 文件**），加上一些元数据文件

这些文件驻留在按时间组织的目录中。这些目录称为**数据桶**。请参阅“Splunk Enterprise 如何存储索引”。

Splunk Enterprise 对其索引进行管理，使搜索更灵活，数据检索更快速，最后按照用户可配置计划对索引归档。Splunk Enterprise 以平面文件形式处理所有内容；不要求任何第三方数据库软件在后台运行。

要开始创建索引，您只需要指定希望 Splunk Enterprise 创建索引的数据导入。您可在任何时候添加更多导入，同时 Splunk Enterprise 也会开始为它们创建索引。请参阅*《数据导入手册》*中的“Splunk Enterprise 可为哪些内容创建索引”，了解如何添加数据导入。

默认情况下，Splunk Enterprise 将所有用户数据放置到一个预先配置的索引中。它还使用多个其他索引完成内部任务。您可以添加新索引以及管理现有索引来满足自己的数据需求。请参阅“管理索引”。

事件处理

索引期间，Splunk Enterprise 会执行**事件处理**。Splunk Enterprise 会处理传入数据，以启用快速搜索和分析，从而以事件形式将结果存储在索引中。创建索引时，Splunk Enterprise 将以各种不同方式增强数据，包括：

- 将数据流分为单个可搜索事件。
- 创建或标识时间戳。
- 提取字段，如主机、数据来源和来源类型。
- 对传入数据执行用户定义的操作，如标识自定义字段、以掩码显示敏感数据、编写新键或修改的键、对多行事件应用换行规则、筛选出不需要的事件以及将事件路由到指定索引或服务器。

*数据导入*还介绍了如何配置事件处理，以满足数据需求。请参阅“事件处理概述”。

索引类型

Splunk Enterprise 支持两种索引：

- 事件索引。事件索引设定的是最小的结构，可以容纳各类数据，包括指标数据。事件索引是默认的索引类型。
- 指标索引。指标索引使用高度结构化的格式以处理更大的指标数据量，并满足低延迟需求。与将指标数据放入事件索引中相比，将相同的数据放入指标索引中会提高性能，降低索引存储的使用频率。有关指标格式的信息，请参阅*指标*手册。

索引器处理方式和管理两种索引类型之间区别很小。尽管事件索引和指标索引的名称、事件处理发生的顺序相同。指标数据实际上只是一种高度结构化的事件数据。

索引器

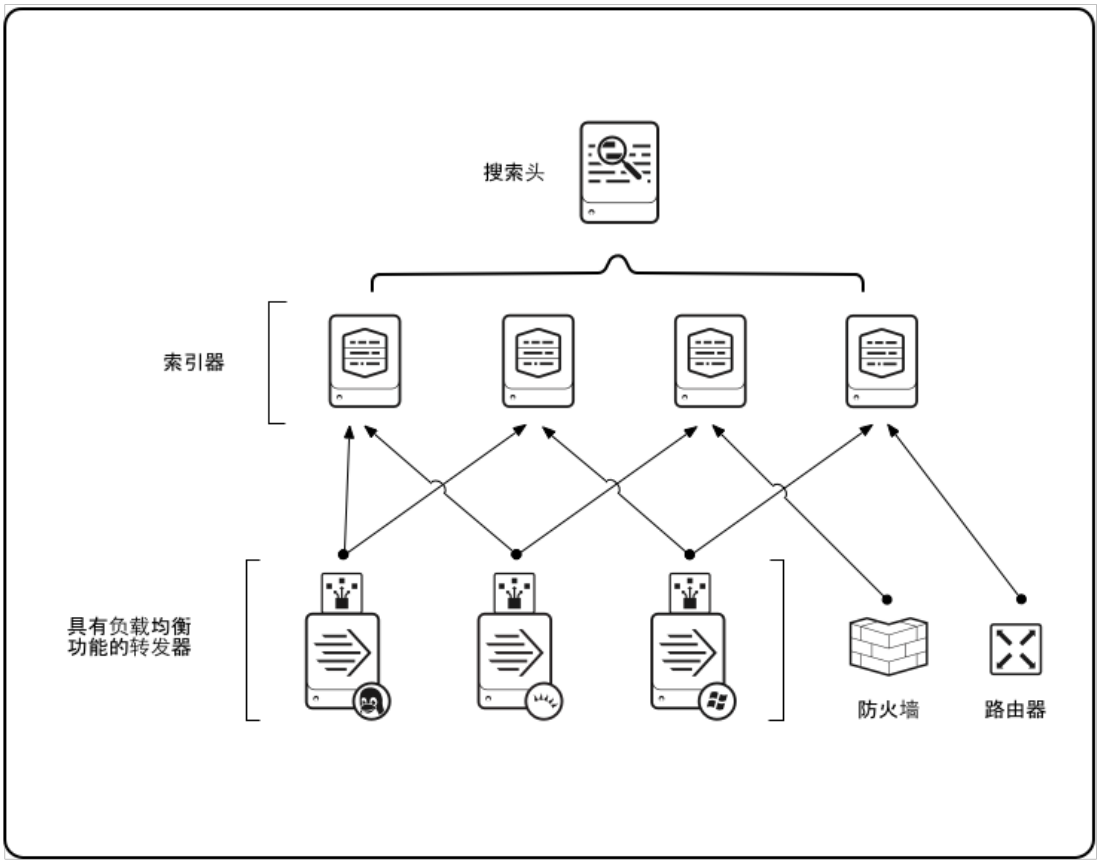
索引器是用于创建和管理索引的 Splunk Enterprise 组件。索引器的主要功能是：

- 为传入数据创建索引。

- 搜索索引数据。

在仅包含一个 Splunk Enterprise 实例的单一计算机部署中，索引器还将处理**数据导入**和**搜索管理**功能。这种类型的小型部署可解决组织单个部门的需求。

对于大规模需求，建立索引功能将从数据导入功能拆分出来，有时也从搜索管理功能拆分出来。在这些较大型的分布式部署中，Splunk Enterprise 索引器可能驻留在自己的计算机上，仅处理建立索引以及其索引数据的搜索。在这些情况下，其他 Splunk Enterprise 组件会接管非索引角色。**转发器**获取数据；索引器为数据建立索引并搜索数据；**搜索头**协调一组索引器中的搜索。下面是一个横向扩展部署的示例：



更多关于在分布式部署中使用索引器的信息，请参阅“分布式部署中的索引器”。

Splunk 索引器只是一个 Splunk Enterprise 实例。要了解如何安装 Splunk Enterprise 实例，请参阅*《安装手册》*。

索引器群集

索引器群集是一组协同工作的 Splunk Enterprise 节点，提供冗余索引和搜索操作。群集中的节点有以下三种类型：

- 单个**管理器节点**，用于管理群集。管理器节点是一种特殊类型的索引器。
- 多个**对等节点**，用于处理群集的索引功能、维护数据的多个副本及为其建立索引，以及对数据执行搜索。
- 一个或多个搜索头，用于协调所有对等节点的搜索。

索引器群集功能会自动从一个对等节点故障转移到下一个对等节点。这意味着，如果一个或多个对等节点出现故障，可继续为传入数据创建索引，且索引数据继续保持可搜索状态。

本手册的第一部分包括适用于所有索引器的配置和管理信息，无论它们是否为群集的一部分。本手册的第二部分从“关于索引器群集和索引复制”主题开始，仅适用于群集。

索引如何工作

Splunk Enterprise 可为任意类型的时间系列数据（具有**时间戳**的数据）**创建索引**。Splunk Enterprise 为数据创建索引时，它会基于时间戳将数据分为**多个事件**。

索引过程遵循事件索引和指标索引的相同步骤序列。

事件处理和数据管道

数据进入索引器，并继续通过执行**事件处理**的管道。最后，将处理过的数据写入磁盘。该管道由几条串连在一起的较短的管道组成。端到端数据管道的单个实例称为**管道集**。

事件处理将出现在两个主要阶段：分析和创建索引。传入 Splunk Enterprise 的所有数据将以大数据块（10,000 个字节）的形式通过**分析管道**进入。分析期间，Splunk Enterprise 会将这些数据块分为若干事件，并将这些事件传递到执行最终处理的**索引管道**。

分析时，Splunk Enterprise 将执行大量操作，其中包括：

- 为每个事件提取一组默认字段，包括 host、source 和 sourcetype。
- 配置字符集编码。
- 使用换行规则识别行尾。虽然大多数事件较短，只占用一两行，但有些事件会很长。
- 标识时间戳，如果时间戳不存在，创建时间戳。Splunk 在处理时间戳的同时会识别事件界限。
- 在此阶段，您可将 Splunk 设置为以掩码显示敏感事件数据（如信用卡号或社会保险号）。也可将其配置为对传入事件应用自定义元数据。

在索引管道中，Splunk Enterprise 会执行其他处理，其中包括：

- 将所有事件分段，然后基于段执行搜索。您可以确定分段的级别，它将影响索引和搜索速度、搜索功能以及磁盘压缩效率。
- 构建索引数据结构。
- 将原始数据和索引文件写入磁盘，其中将执行后索引压缩。

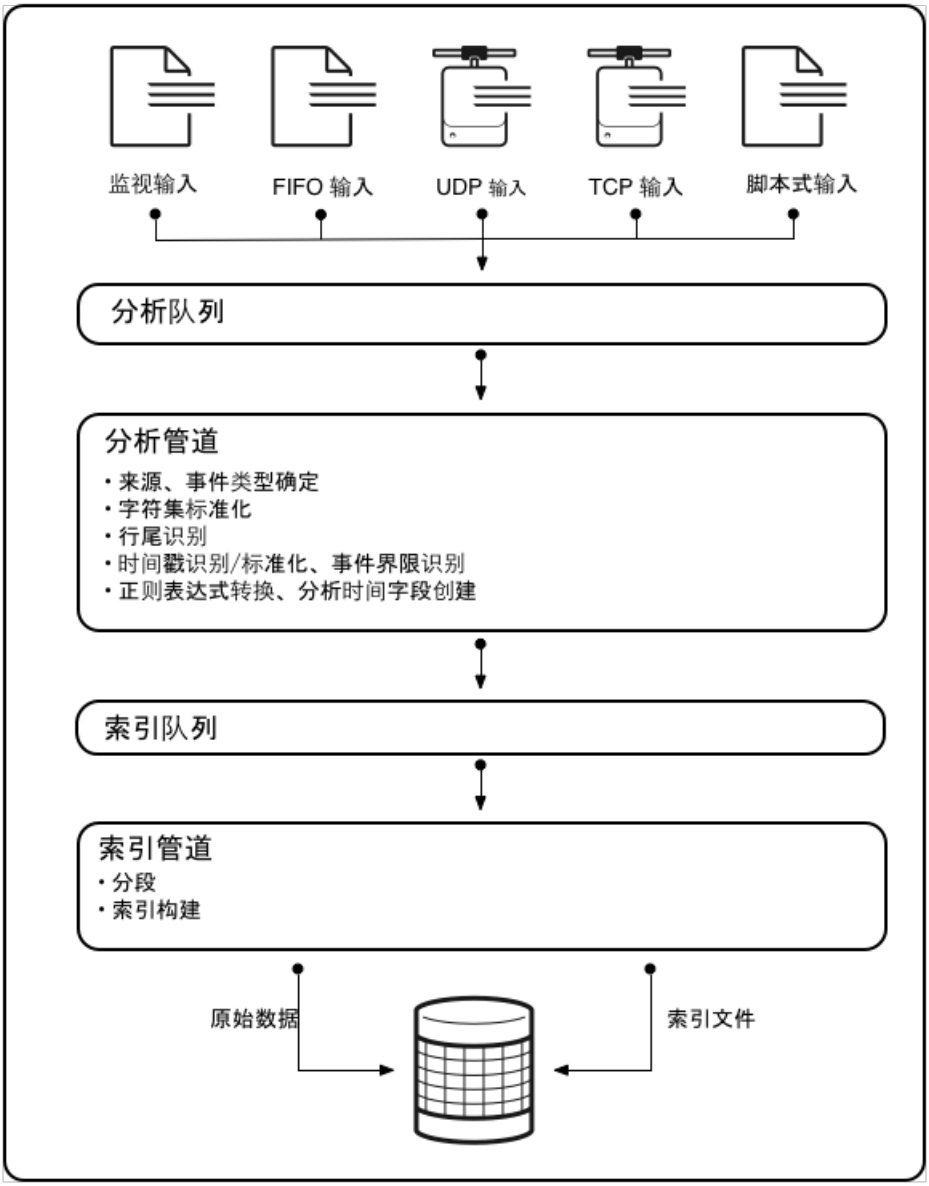
注意：“索引”一词更常用于指整个事件处理过程，包括分析管道和索引管道。分析与索引管道间的差异主要与部署**重型转发器**的时间相关。

重型转发器可以通过分析管道来处理原始数据，然后将分析后的数据转发到索引器上执行最终的索引建立。**通用转发器**不会以这种方式分析数据。反之，通用转发器将原始数据转发给索引器，然后索引器同时通过两种管道来处理该数据。

注意：这两种类型的转发器一定会在某些结构化数据上执行一种分析操作。请参阅*《数据导入手册》*中的“从具有标头的文件中提取数据”。

更多关于事件以及索引器如何将数据转换为事件的信息，请参阅*《数据导入手册》*中的“配置事件处理”一章。

下图显示索引创建过程中固有的主要进程：



注意：此图是索引架构的简化视图。它提供了架构的功能视图，而且没有详细介绍 Splunk Enterprise 内部过程。特别的是，分析管道实际上是由三个管道组成：**分析**、**合并**和**键入**，这三个管道共同处理分析功能。故障排除期间，该区别是有意义的，但通常不会影响您配置或部署 Splunk Enterprise 的方式。

关于数据管道及数据管道如何影响部署决策的详细介绍，请参阅*《分布式部署手册》*中的“数据如何通过 Splunk Enterprise：数据管道”。

索引中的内容

Splunk Enterprise 将其处理的数据存储在索引中。索引由子目录的集合组成，被称为**数据桶**。数据桶主要由两种类型的文件组成：**元数据文件**和**索引文件**。请参阅“Splunk Enterprise 如何存储索引”。

索引数据的不可变性

将数据添加到索引之后，您无法编辑或以其他方式更改数据。您可以删除索引中的所有数据，或者根据策略删除单个索引数据桶或选择进行归档，但是您不能选择性地删除存储中的单个事件。

请参阅“删除索引和索引的数据”。

默认索引集

Splunk Enterprise 随附大量预先配置的索引，包括：

- **main**：这是默认 Splunk Enterprise 索引。如果没有另外指定，所有处理后数据均存储在此处。

- **_internal**：存储 Splunk Enterprise 内部日志和处理指标。
- **_audit**：包含与文件系统更改监视器、审计和所有用户搜索历史相关的事件。

Splunk Enterprise 管理员可以创建新索引、编辑索引属性、删除不需要的索引以及重新定位现有索引。Splunk Enterprise 管理员通过“Splunk Web”、CLI 和配置文件（如 indexes.conf）管理索引。请参阅“管理索引”。

索引时间对比搜索时间

Splunk Enterprise 文档中包含对术语**“索引时间”**和**“搜索时间”**的引用。这两个术语用于区分创建索引期间发生的处理的类型以及运行搜索时发生的类型。

在管理 Splunk Enterprise 时考虑这一区别非常重要。例如，假设您想要使用自定义**来源类型**和**主机**。您应该在开始创建索引之前定义这些自定义来源类型和主机，这样才能在索引创建过程中用它们来标记事件。创建完索引后，您将无法再更改主机或来源类型分配。

如果您到开始为数据创建索引时才发现自己忘记创建自定义来源类型和主机，您只有两种选择：（1）重新为数据创建索引，这样才能把自定义来源类型和主机应用到现有数据和新的数据；或者（2）通过使用替代值标记事件管理搜索时间的问题。

相反，通常情况下，最好在搜索时间执行大部分知识构建活动，如**字段提取**。索引时间自定义字段提取可以在索引时间和搜索时间降低性能。在创建索引期间增加提取的字段数时，将减慢创建索引进程的速度。以后，对索引的搜索速度也会减慢，因为索引已被更多字段增大，对较大的索引进行搜索需要的时间会长一些。

通过改为基于搜索时间字段提取，您可以避免此类性能问题。关于搜索时间字段提取的详细信息，请参阅*《知识管理器手册》*中的“关于字段”和“当 Splunk Enterprise 提取字段时”。

索引时间期间

索引时间进程发生于两个时间点之间，分别为使用数据时和将数据写入磁盘时。

下列进程发生于索引时间期间内：

- 默认字段提取（如 host、source、sourcetype 和 timestamp）
- 针对特定输入的静态或动态主机分配
- 默认主机分配覆盖
- 来源类型自定义
- 自定义索引时间字段提取
- 结构化数据字段提取
- 事件时间戳
- 事件换行
- 事件分段（同时也在搜索时间发生）

搜索时间期间

搜索时间进程在运行搜索的同时发生，因为事件通过搜索来搜集。在搜索时间将发生以下进程：

- 事件分段（同时也在索引时间发生）
- 事件类型匹配
- 搜索时间字段提取（自动和自定义字段提取，包括多值字段和已计算字段）
- 字段别名
- 其他字段的查找
- 来源类型重命名
- 标记

数据管道

数据管道提供一种更为详细的方式，供您了解数据通过系统的进度。数据管道对帮助用户了解如何在分布式部署中分配配置和工作十分有用。请参阅*《分布式部署手册》*中的“数据如何通过 Splunk：数据管道”。

安装索引器

默认情况下，所有完整 Splunk Enterprise 实例作为索引器。要了解如何安装 Splunk Enterprise 实例，请参阅《安装手册》。然后，返回本手册以了解如何配置索引器。

如果您计划在分布式部署中部署索引器，阅读下一主题“分布式环境中的索引器”。

分布式部署中的索引器

重要提示：为更好地理解本主题，您应熟悉*《分布式部署手册》*中介绍的 Splunk Enterprise 分布式环境。

索引器是用于创建和管理索引的 Splunk Enterprise 组件。索引器的主要功能是：

- 为传入数据创建索引。
- 搜索索引数据。

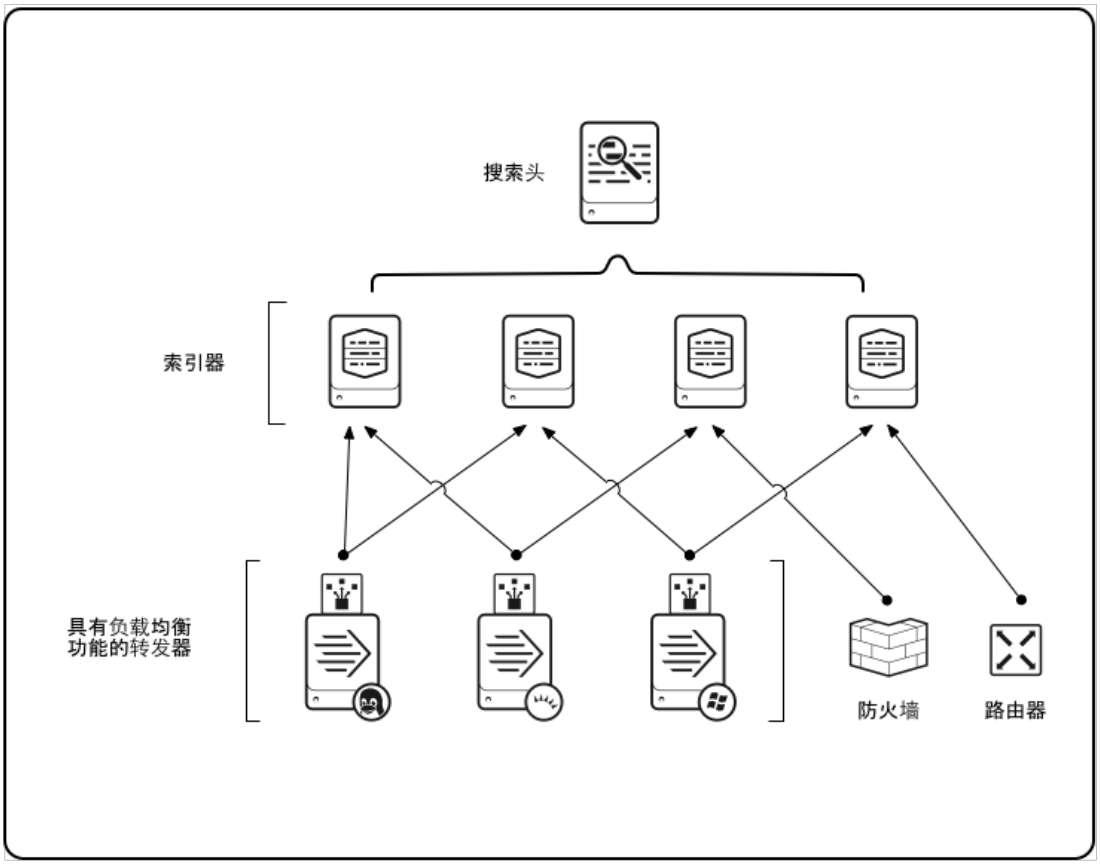
在仅包含一个 Splunk Enterprise 实例的单一计算机部署中，索引器还将处理**数据导入**和**搜索管理**功能。

对于大规模需求，建立索引功能将从数据导入功能拆分出来，有时也从搜索管理功能拆分出来。在这些较大型的分布式部署中，索引器可能驻留在自己的计算机上，仅处理建立索引以及其索引数据的搜索。在这些情况下，其他 Splunk Enterprise 组件会接管非索引角色。

例如，您可能有一组用于生成事件的 Windows 和 Linux 计算机，它们需要转到中央索引器进行合并。通常，实现此操作的最佳方式是在每一个生成事件的计算机上安装一个轻型 Splunk Enterprise 实例，称为**转发器**。这些转发器处理数据导入，并通过网络将数据发送到驻留在自己计算机上的索引器。

同样地，如果您有大量已索引的数据并有众多并发用户要对这些数据执行搜索，明智的做法是将搜索管理功能与索引创建拆分开来。在这种情形下（称为**分布式搜索**），一个或多个**搜索头**会将搜索请求分布到多个索引器。这些索引器仍执行自己索引的实际搜索，但搜索头会跨所有索引器管理整个搜索进程，然后将合并后的搜索结果提供给用户。

下面是横向扩展部署的一个示例：



对于分布式部署，虽然索引创建和事件处理的基本问题保持不变，但需要在计划索引策略时考虑部署需求，这一点非常重要。

将数据转发到索引器

要将远程数据转发到索引器，请使用转发器，它们是 Splunk Enterprise 实例，用于接收数据导入、合并，再将数据发送到 Splunk Enterprise 索引器。转发器有以下两种：

- **通用转发器**。它们会占用主机的一小块空间。该转发器用于在将传入数据流转发到索引器（也称为**接收器**）之前，对传入数据流执行最低程度的处理。
- **重型转发器**。它们保留完整 Splunk Enterprise 实例的大多数功能。在将数据转发到接收索引器之前，它们能够分析数据。（关于分析与索引之间的区别，请参阅“索引如何工作”。）它们能够在本地存储已索引的数据，还能够将分析后的数据转发到接收器以便在该计算机上执行最终的索引创建。

在将数据转发到索引器之前，这两种类型的转发器均使用元数据（如主机、数据来源和来源类型）标记数据。

在处理来自远程来源的大量数据或不同类型的数据时，使用转发器可有效利用资源。通过提供**负载均衡**、数据**过滤**和**路由**功能，它们还将启用许多感兴趣的部署拓扑结构。

关于转发器的深入介绍，包括配置和详细使用案例，请参阅*《转发数据手册》*。

跨多个索引器搜索

在分布式搜索中，搜索头将搜索请求发送到索引器，然后将合并后的结果返回给用户。这对于实现横向扩展、访问控制和管理地理分散的数据而言将非常有用。

关于分布式搜索和搜索头的深入介绍，包括配置和详细使用案例，请参阅*《分布式搜索手册》*。

索引器群集还使用搜索头在群集的对等节点中协调搜索。请参阅“关于索引器群集和索引复制”。

在分布式环境中部署索引器

要使部署的分布式环境与本主题较早介绍的图形类似，您需要安装和配置三种类型的组件：

- 索引器
- 转发器（通常为通用转发器）
- 搜索头

安装和配置索引器

默认情况下，所有完整 Splunk Enterprise 实例作为索引器。对于横向扩展，您可以在单独计算机上安装多个索引器。

要了解如何安装 Splunk Enterprise 实例，请参阅*《安装手册》*。

然后，返回本手册以了解有关配置每个单独索引器以满足特定部署需求的信息。

安装和配置转发器

典型分布式部署具有大量转发器，提供数据给几个索引器。对于大部分转发用途，通用转发器是最佳选择。通用转发器是独立于完整 Splunk Enterprise 实例的单独可下载程序。

要了解如何安装和配置转发器，请参阅*《转发数据》*。

安装和配置搜索头

您可以安装一个或多个搜索头以处理分布式搜索需求。搜索头仅是专门配置的完整 Splunk Enterprise 实例。

要了解如何配置搜索头，请参阅*《分布式搜索》*。

其他部署任务

您需要指定**许可证主服务器**以配置 Splunk Enterprise 许可。更多信息请参阅*《管理员手册》*中的“配置 Splunk Enterprise 许可证”章节。

您还可以使用 Splunk Enterprise **部署服务器**简化更新部署组件这一任务。关于如何配置部署服务器的详细信息，请参阅*《更新 Splunk Enterprise 实例》*。

安装索引器群集

如果数据可用性、数据保真度和数据恢复是您部署的主要问题，则应考虑部署索引器群集而不是一系列单个索引器。更多信息请参阅“关于索引器群集和索引复制”。

管理索引

关于管理索引

添加数据时，索引器将对它进行处理并将其存储在**索引**中。默认情况下，您提供给索引器的数据会存储在 **main** 索引中，但可创建及指定其他索引，以用于其他数据导入。

索引是目录和文件的集合。它们位于 $SPLUNK_HOME/var/lib/splunk 下方。索引目录也称为**数据桶**，按时间进行组织。有关索引存储的相关信息，请参阅“Splunk Enterprise 如何存储索引”。

除了 **main** 索引，Splunk Enterprise 还附带了许多预先配置的内部索引。对内部索引命名时，前面加上一个下划线 (_)。要查看 Splunk Web 中的完整索引列表，单击 Splunk Web 上半部分中的**设置**链接，然后选择**索引**。列表包括：

- **main**：默认 Splunk Enterprise 索引。如果没有另外指定，所有处理的外部数据均存储在此处。
- **_internal**：此索引包括 Splunk Enterprise 内部日志。
- **_metrics**：此索引包含 Splunk Enterprise 内部数据，该数据以指标数据点形式存储。
- **_audit**：来自文件系统更改监视器、审计和所有用户搜索历史的事件。
- **_introspection**：此索引提供有关 Splunk Enterprise 实例和环境的数据。

本手册中的多个主题都介绍了索引的管理方式。特别的是，以下主题在索引管理方面将会为您提供很大帮助：

- 创建自定义索引
- 从 Splunk 删除索引和数据
- 配置索引存储
- 移动索引数据库
- 对索引数据使用多个分区
- 配置索引大小
- 对磁盘使用情况设置限制
- 备份索引数据
- 设置退休和归档策略

索引类型

Splunk Enterprise 支持两种索引：

- 事件索引。事件索引设定的是最小的结构，可以容纳各类数据，包括指标数据。事件索引是默认的索引类型。
- 指标索引。指标索引使用高度结构化的格式以处理更大的指标数据量，并满足低延迟需求。与将指标数据放入事件索引中相比，将相同的数据放入指标索引中会提高性能，降低索引存储的使用频率。有关指标格式的信息，请参阅*指标*手册。

索引器处理两种索引类型方式的区别很小。如有任何区别，都会在相关主题中有所介绍。

有关索引创建过程的详细信息

要了解更多索引相关信息，请参阅：

- 本手册中的“索引如何工作”主题。
- 本手册中的“Splunk 如何存储索引”主题。
- 本手册中的“关于群集和索引复制”一章。
- *《数据导入手册》*中的“配置事件处理”一章。
- 有关使用巨大数据集的信息，请参阅*《知识管理器手册》*中的“设置和使用摘要索引”一章。

创建自定义索引

您可以创建两种索引：

- 事件索引
- 指标索引

事件索引是默认的索引类型。要新建事件索引，请参阅“新建事件索引”。

要创建指标索引，请参阅“创建指标索引”。有关指标索引的常规信息，请参阅*指标*手册，从“指标概述”开始阅读。

有关新建 SmartStore 索引的信息，请参阅“添加 SmartStore 索引”。

新建事件索引

默认情况下，main 索引保存您的所有事件。该索引还相当于所有未指定索引的输入或搜索命令的默认索引，但可以更改默认索引。

具有 Splunk Enterprise 许可证，您就能添加无限多个其他索引。可使用 Splunk Web、CLI 或 indexes.conf 添加索引。

本主题涉及：

- 需要多个索引的原因。
- 如何创建新索引。
- 如何将事件发送到特定索引。
- 如何搜索特定索引。

为何需要多个索引？

需要多个索引有以下几大原因：

- 用于控制用户访问权限。
- 用于适应不同的保存策略。
- 用于提高特定情形下的搜索速度。

设置多个索引的主要原因是控制用户对索引中数据的访问权限。当您为用户分配**角色**时，可基于用户角色来限制用户对特定索引的搜索。

此外，如果您对于保存不同的数据集拥有不同的策略，则可能需要将数据发送到不同的索引，然后为每个索引设置不同的存档或保存策略。

设置多个索引的另一个原因与搜索的工作方式相关。如果您同时将大容量/高噪音数据来源和小容量数据来源提供给同一个索引，并且您主要从小容量数据来源中搜索事件，则搜索速度将会慢于所需速度，因为索引器还要必须在大容量数据来源的所有数据中搜索。为使这种情况得到缓解，可以为每个数据来源创建专用索引并将数据从每个来源发送到其专用索引。然后，可以指定对哪个索引执行搜索。您可能会发现搜索速度已经增加。

新建事件索引

可以通过 Splunk Web、CLI 或直接编辑 indexes.conf 来新建事件索引。

注意：要将新索引添加到索引器群集，必须直接编辑 indexes.conf，不能通过 Splunk Web 或 CLI 添加索引。关于如何配置群集 indexes.conf 的信息，请参阅“在索引器群集中配置对等节点索引”。该主题中包含了创建新群集索引的示例。

使用 Splunk Web

您不能使用 Splunk Web 添加 SmartStore 索引。如果索引器有 SmartStore 索引，您也不能使用 Splunk Web 添加非 SmartStore 索引。

1. 在 Splunk Web 中，导航到**设置 > 索引**，然后单击**新建**。
2. 要创建新索引，输入：
    - 索引的名称。用户定义的索引名称只能由数字、小写字母、下划线和连字符组成。它们不能以下划线或连字符开始，或包含单词 "kvstore"。
    - 索引数据类型。关于事件数据，请单击**事件**。这是默认的数据类型。
    - 索引数据存储的路径位置：
        - 主路径。保留空白，以使用默认值 $SPLUNK_DB/<index_name>/db
        - 冷路径。保留空白，以使用默认值 $SPLUNK_DB/<index_name>/colddb
        - 解冻路径。保留空白，以使用默认值 $SPLUNK_DB/<index_name>/thaweddb
    - 启用/禁用数据完整性检查。
    - 整个索引的最大大小。默认为 500,000MB。
    - 每个索引数据桶的最大大小。设置最大大小时，对大容量索引（如 main 索引）使用 auto_high_volume，否则使用 auto。
    - 冻结归档路径。如果希望归档冻结数据桶，请设置此字段。有关数据桶归档的信息，请参阅“归档索引的数据”。
    - 索引驻留在的应用。
    - tsidx 保存策略。请参阅“减少 tsidx 使用”。

    关于索引设置的更多信息，请参阅“配置索引存储”。
3. 单击**保存**。

您可以在 Splunk Web 上，单击**设置**菜单的**索引**部分的索引名称来编辑索引。Splunk Web 中您不能更改的属性将显示为灰色。要更改这些属性，需要编辑 indexes.conf，然后重新启动索引器。

注意：某些索引属性只能通过编辑 indexes.conf 文件进行配置。请参阅 indexes.conf 主题获得完整的属性列表。

使用 CLI

导航到 $SPLUNK_HOME/bin/ 目录，然后使用 add index 命令。您不需要先停止索引器。

要添加名为 "fflanda" 的新索引，请输入以下命令：

```
splunk add index fflanda
```

注意：用户定义的索引名称只能由数字、小写字母、下划线和连字符组成。它们不能以下划线或连字符开始，或包含单词 "kvstore"。

如果不希望新索引使用默认路径，可以使用参数来指定新位置：

```
splunk add index foo -homePath /your/path/foo/db -coldPath /your/path/foo/colddb
    -thawedPath /your/path/foo/thawedDb
```

您还可以通过 CLI 编辑索引的属性。例如，要使用 CLI 编辑名为 "fflanda" 的索引，键入：

```
splunk edit index fflanda -<parameter><value>
```

关于索引设置的详细信息，请参阅“配置索引存储”。

编辑 indexes.conf

要添加新索引，请在 indexes.conf（位于 $SPLUNK_HOME/etc/system/local）中添加一个段落，该段落由新索引名称进行标识。例如：

```
[newindex]
homePath=<path for hot and warm buckets>
coldPath=<path for cold buckets>
thawedPath=<path for thawed buckets>
...
```

关于索引设置的信息，请参阅“配置索引存储和 indexes.conf 规范文件”。

注意：用户定义的索引名称只能由数字、小写字母、下划线和连字符组成。它们不能以下划线或连字符开始，或包含单词 "kvstore"。

编辑 indexes.conf 后必须重新启动索引器。

关于在群集节点上添加或编辑索引配置的信息，请参阅“在索引器群集中配置对等节点索引”。

将事件发送到特定索引

默认情况下，所有外部事件会转到名为 **main** 的索引。但是，您可能希望将某些事件发送到其他索引。例如，您可能希望将来自特定输入的所有数据发送到其自己的索引。或者，您可能希望对数据进行分段或将来自某噪音数据来源的事件数据发送到专用于接收该数据的索引。

重要提示：要将事件发送到某特定索引，该索引必须已经存在于索引器上。如果将任何事件发送到并不存在的索引，那么索引器将丢弃这些事件。

将来自某数据导入的所有事件发送到特定索引

要将来自某特定数据导入的所有事件发送到特定索引，请在数据进入系统所用的 Splunk Enterprise 组件（索引器本身或将数据发送到索引器的转发器）上的 inputs.conf 文件的输入段落中添加以下行：

```
index = <index_name>
```

以下示例 inputs.conf 段落将来自 /var/log 的所有数据发送到名为 fflanda 的索引：

```
[monitor:///var/log]
disabled = false
index = fflanda
```

将特定事件发送到不同索引

正如您可以将事件发送到特定队列一样，您也可以将特定事件发送到特定索引。可以在索引器自身中配置，而*不是*在将数据发送到索引器的转发器（如果有）中配置。

要将特定事件发送到特定索引，请在索引器上编辑 props.conf 和 transforms.conf：

1. 确定一个可用于区分各个事件的通用属性。

2. 在 props.conf 中，为数据来源、来源类型或主机创建一个段落。此段落指定 transforms_name，对应于您将在 transforms.conf 中创建的包含正则表达式的段落。

3. 在 transforms.conf 中，创建一个以您在步骤 2 中指定的 transforms_name 进行命名的段落。此段落：

- 指定一个与在步骤 1 中确定的属性相匹配的正则表达式。
- 指定应将与属性匹配的事件发送到的替代索引。

以下各部分将对步骤 2 和 3 中的详细信息进行填写。

编辑 props.conf

将以下段落添加到 $SPLUNK_HOME/etc/system/local/props.conf：

```
[<spec>]
TRANSFORMS-<class_name> = <transforms_name>
```

请注意以下事项：

- <spec> 是以下各项中的其中一项：
  - <sourcetype>，事件的来源类型
  - host::<host>，其中 <host> 为事件所在的主机
  - source::<source>，其中 <source> 为事件的来源
- <class_name> 是唯一标识符。
- <transforms_name> 是要为 transforms.conf 中的转换指定的唯一标识符。

编辑 transforms.conf

将以下段落添加到 $SPLUNK_HOME/etc/system/local/transforms.conf：

```
[<transforms_name>]
REGEX = <your_custom_regex>
DEST_KEY = _MetaData:Index
FORMAT = <alternate_index_name>
```

请注意以下事项：

- <transforms_name> 必须与 <transforms_name> 标识符（您在 props.conf 中指定的）匹配。
- <your_custom_regex> 必须为您之前在步骤 1 中确定的属性提供一个匹配。
- DEST_KEY 必须设置为索引属性 _MetaData:Index。
- <alternate_index_name> 指定要将事件发送到的替代索引。

示例

此示例将根据事件的日志类型将 windows_snare_log 来源类型的事件发送到相应的索引。"Application" 日志将被发送到替代索引，而所有其他日志类型（如 "Security"）将被发送到默认索引。

为了对此进行确定，将使用 props.conf 对 windows_snare_log 来源类型的事件进行定向。具体做法是通过名为 "AppRedirect" 的 transforms.conf 段落，其中的正则表达式会对 "Application" 日志类型进行查找。在相应位置中 "Application" 匹配的任何事件都会被发送到替代索引 "applogindex"。所有其他事件会转到默认索引。

1. 确定属性

本示例中的事件如下所示：

```
web1.example.com        MSWinEventLog        1        Application        721        Wed Sep 06 17:05:31 2006
4156        MSDTC        Unknown User        N /A        Information        WEB1        Printers                String
message: Session idle timeout over, tearing down the session.        179
```

```
web1.example.com        MSWinEventLog        1        Security        722       Wed Sep 06 17:59:08 2006
576        Security        SYSTEM        User        Success Audit        WEB1        Privilege Use
Special privileges assigned to new logon:     User Name:      Domain:      Logon
ID: (0x0,0x4F3C5880)     Assigned: SeBackupPrivilege   SeRestorePrivilege
SeDebugPrivilege   SeChangeNotifyPrivilege   SeAssignPrimaryTokenPrivilege 525
```

某些事件包含 "Application" 值，而其他事件在相同的位置包含 "Security" 值。

2. 编辑 `props.conf`

将此段落添加到 `$SPLUNK_HOME/etc/system/local/props.conf`：

```
[windows_snare_syslog]
TRANSFORMS-index = AppRedirect
```

此段落将 `windows_snare_syslog` 来源类型的事件定向到 `AppRedirect` 段落（位于 `transforms.conf`）。

3. 编辑 `transforms.conf`

将此段落添加到 `$SPLUNK_HOME/etc/system/local/transforms.conf`：

```
[AppRedirect]
REGEX = MSWinEventLog\s+\d+\s+Application
DEST_KEY = _MetaData:Index
FORMAT = applogindex
```

此段落将处理在此处由 `props.conf` 定向的事件。与正则表达式匹配的事件（因为它们在指定位置包含 "Application" 字符串）将被发送到替代索引 "applogindex"。所有其他事件会被正常发送到默认索引。

搜索特定索引

除非搜索显式指定一个索引，否则索引器在搜索时，会将默认索引（默认为 **main**）作为目标。例如，下面的搜索命令会在 `hatch` 索引中进行搜索：

```
index=hatch userid=henry.gale
```

创建或编辑给定的角色时，也可以该角色指定要搜索的替代默认索引。

创建指标索引

您可以通过 Splunk Web、CLI、REST API 或直接编辑 `indexes.conf` 来创建指标索引。有关指标的更多信息，请参阅*指标*手册中的指标概览。

使用 Splunk Web

1. 在 Splunk Web 中，导航到**设置 > 索引**，然后单击**新建**。
2. 请输入索引名称创建**索引名称**。用户定义的索引名称只能由数字、小写字母、下划线和连字符组成。索引名称不能以下划线或连字符开始，或包含单词 "kvstore"。
3. 请单击**指标**选择**索引数据类型**。
4. （可选）如果希望指标索引以增加后的粒度级别存储指标数据点，请将**时间戳分辨率**设置为**毫秒**。具有毫秒级时间戳分辨率的指标索引降低了搜索性能。请参阅“带毫秒级时间戳的指标索引”。
5. 必要时，请输入索引的剩余属性。详细信息，请参阅新建事件索引。
6. 单击**保存**。

使用命令行界面（CLI）

1. 打开命令提示符。
2. 导航到 **$SPLUNK_HOME/bin/** 目录。
3. 使用 `add index` 命令创建索引。用户定义的索引名称只能由数字、小写字母、下划线和连字符组成。索引名称不能以下划线或连字符开始，或包含单词 "kvstore"。

例如，要创建名为 `mymetricsindex` 的索引，请输入以下命令：

```
splunk add index mymetricsindex -datatype metric
```

要列出所有指标索引，请输入以下命令：

```
splunk list index -datatype metric
```

要列出所有索引（包括事件索引），请输入以下命令：

```
splunk list index -datatype all
```

使用 REST API

使用 **/data/indexes** 端点，通过 "datatype=metric" 参数创建索引。详细信息，请参阅*《REST API 参考手册》*中的 **/data/indexes**。

例如，要创建名为 mymetricsindex 的指标索引，请输入以下命令：

```
curl -k -u admin:pass https://localhost:8089/services/data/indexes  \
    -d name=mymetricsindex                                          \
    -d datatype=metric
```

要使用 REST API 列出所有指标索引，请输入以下命令：

```
curl -k -u admin:pass https://localhost:8089/services/data/indexes?datatype=metric
```

要列出所有索引（包括事件索引），请输入以下命令：

```
curl -k -u admin:pass https://localhost:8089/services/data/indexes?datatype=all
```

编辑 indexes.conf

要创建新的指标索引，请在 indexes.conf（位于 $SPLUNK_HOME/etc/system/local）中添加一个段落，该段落由新索引名称进行标识。更改 datatype 参数为 datatype = metric。

例如，要创建名为 "mymetricsindex" 的指标值索引，请添加以下段落：

```
[mymetricsindex]
homePath=<path for hot and warm buckets>
coldPath=<path for cold buckets>
thawedPath=<path for thawed buckets>
datatype = metric
metric.timestampResolution = <s | ms>
...
```

关于索引设置的信息，请参阅“配置索引存储和 indexes.conf 规范文件”。

用户定义的索引名称只能由数字、小写字母、下划线和连字符组成。它们不能以下划线或连字符开始，或包含单词 "kvstore"。

编辑 indexes.conf 后必须重新启动索引器。

关于在群集节点上添加或编辑索引配置的信息，请参阅“在索引器群集中配置对等节点索引”。

带毫秒级时间戳的指标索引

默认情况下，只能以秒级精度搜索指标索引。这和事件索引不同，事件索引默认情况下可以亚秒级精度进行搜索。

如果您要处理大量的指标数据源，例如每秒可能产生数百万个指标数据点的公用电网，则意味着指标索引中填充了样本指标数据点或指标数据点，这些数据点是原始指标数据的汇总视图，且以固定间隔获取。

如果您担心索引量太高，这可能是一件好事。具有秒级精度的指标索引可以使索引保持精简，并让您不必在相对较短的时间范围内搜索大量事件。但这也意味着您无法运行具有亚秒级精度的基于时间的指标搜索。同样，您无法设置按亚秒 span 值分组的 mstats 搜索。

如果您需要能够以亚秒级精度执行指标搜索，请为新的指标索引指定**毫秒级**的**时间戳分辨率**。与具有默认秒级时间戳精度的指标索引相比，具有毫秒级时间戳分辨率的指标索引可能会降低搜索性能。

与设置为秒级精度的类似指标相比，设置为毫秒级精度的指标可能会导致更高的许可证使用率。每个指标数据点的许可成本保持不变，但是与从同一源中提取数据的秒级精度索引相比，毫秒级精度索引可以索引更多数据点。

关于更改指标索引的时间戳分辨率

创建指标索引后，可以更改其分辨率。但是，如果将指标索引的时间戳分辨率从毫秒更改为秒，则对于定期针对该指标索引运行搜索的人员来说，可能情况看起来像是数据丢失。这是因为索引在更改后不会以毫秒级分辨率引入数据。

当索引处于毫秒级时间戳分辨率时，索引的指标数据点可能也具有同样的时间戳。

_timestamp（单位为秒）
1.000
1.001
1.002
2.000
2.435
3.123
3.651
4.000

四秒钟后，如果将时间戳分辨率从毫秒级时间戳分辨率更改为秒级时间戳分辨率，则索引将被限制为每秒只能为一个指标数据点建立索引：

_timestamp（单位为秒）
5.000
6.000
7.000
8.000
9.000

有些用户可能会将其视为数据丢失，但实际上这些数据只是以较小粒度的时间戳分辨率显示。

同样，从秒级时间戳分辨率切换到毫秒级时间戳分辨率的指标索引的用户可能会惊讶地发现，索引引入的事件比切换之前要多。

作为 Splunk Enterprise 部署的管理员，您有责任传达这个变化以及它对用户的影响。

删除索引和索引的数据

您可以从索引器删除索引数据，甚至是整个索引。下面是主要选项：

- 从后续的搜索中删除事件。
- 从一个或多个索引中删除所有数据。
- 删除或禁用整个索引。
- 基于退休策略删除旧数据。

删除数据是不可撤消的操作。如果使用本主题中介绍的任何方法删除了数据，但想要重新获得此数据，那么您必须对适用的数据源重新建立索引。

从后续的搜索中删除事件

Splunk 搜索语言提供了可从后续的搜索中删除事件数据的命令 `delete`。

`delete` 命令只能用于事件索引。您不能将其与指标索引一起使用

实时搜索过程中无法运行 delete 命令。如果您尝试在实时搜索过程中使用 delete 命令，Splunk Enterprise 将显示错误。

delete 命令只删除后续搜索中的事件。数据本身仍保留在索引中。

什么用户可以执行删除操作？

只有具有 "delete_by_keyword" **功能**的用户可以运行 delete 命令。默认情况下，Splunk Enterprise 随附了 "can_delete" 这个特殊**角色**，该角色拥有此功能（其他用户都没有此功能）。管理员角色在默认情况下没有此功能。建议您创建一个特殊用户，在要删除索引数据时登录此用户。

更多信息请参阅*《确保 Splunk Enterprise 安全手册》*中的“添加和编辑角色”。

如何删除

首先，运行一个搜索，该搜索将返回您想要删除的事件。请确保此搜索*只*会返回您想要删除的事件，而不会返回其他事件。确定之后，可以将此搜索结果通过管道符传递给 delete 命令。

例如，如果想要删除根据名为 /fflanda/incoming/cheese.log 的数据来源建立索引的事件，以便它们不会再出现在搜索中，请执行以下操作：

1. 禁用或删除该数据来源，以便不再为其建立索引。

2. 在您的索引中搜索来自该数据来源的事件：

```
source="/fflanda/incoming/cheese.log"
```

3. 查看结果，确认这是您要删除的数据。

4. 确认这是您要删除的数据后，将此搜索的结果通过管道符传递给 delete：

```
source="/fflanda/incoming/cheese.log" | delete
```

有关更多示例的信息，请参阅《搜索参考手册》中与 delete 命令相关的页面。

注意：在 Windows 上运行 Splunk 时，将示例中的正斜线 (/) 替换为反斜线 (\)。

将搜索通过管道符传递给 delete 命令会对该搜索所返回的所有事件进行标记，以便后续的搜索不会再返回这些事件。搜索时，没有用户（即使具有管理员权限）可以看到此数据。

注意：通过管道符传递给 delete 不会回收磁盘空间。数据实际上并未从索引中删除，只是对搜索不可见。

delete 命令不会更新事件的元数据，因此任何元数据搜索都会仍然包括这些事件，即使它们已不可搜索。**所有索引数据**主仪表板将仍然显示对已删除的数据来源、主机或来源类型的事件计数。

删除操作和索引器群集

在索引复制的一般过程中，delete 操作的影响会快速传输到群集中的所有数据桶副本中，通常仅需几秒或几分钟，具体取决于群集负载、数据总量以及受 delete 操作影响的数据桶。在传输间隔期间，搜索可返回那些已经删除的结果。

同时，如果在传输所有结果之前，一个拥有主要数据桶副本的节点在 delete 操作时出故障，那么一些删除将会丢失。在这种情况下，您必须在来自故障节点的主要副本重新分配后，再次运行此操作。

从一个或所有索引中删除所有数据

要从磁盘中永久删除索引数据，请使用 CLI clean 命令。此命令将完全删除一个或所有索引中的数据，具体取决于您是否提供了 <index_name> 参数。通常，在为所有数据重新建立索引之前，需要运行 clean 命令。

注意：clean 命令不适用于群集索引。

如何使用 clean 命令

以下是使用 clean 命令的主要方法：

- 要访问 clean 的帮助页面，键入：

```
splunk help clean
```

- 要从**所有索引**中永久删除数据，键入：

```
splunk clean eventdata
```

- 要从**单个索引**中永久删除数据，键入：

```
splunk clean eventdata -index <index_name>
```

其中，<index_name> 是目标索引的名称。

- 添加 -f 参数可以强制 clean 跳过其确认提示。

重要提示：运行 clean 命令之前，必须停止索引器。

注意：在 5.0 之前版本的 Splunk Enterprise 中，运行 clean 命令会导致索引器将索引的下一个数据桶 ID 值重置为 0。从 5.0 版本开始，将不再出现这种问题。所以，如果最后数据桶的 ID 为 3，那么在运行 clean 之后，下一个数据桶 ID 将为 4，而不是 0。更多关于数据桶命名约定和数据桶 ID 的信息，请参阅索引目录的结构。

示例

此示例会删除所有索引的数据：

```
splunk stop
splunk clean eventdata
```

以下示例将删除 _internal 索引的数据，并强制 Splunk 跳过确认提示：

```
splunk stop
splunk clean eventdata -index _internal -f
```

删除整个索引

要将索引从非群集索引器中整个（不仅仅是其中包含的数据）删除，您可以使用 Splunk Web 或 CLI。您也可以直接编辑 indexes.conf

删除索引之前，先查看所有 inputs.conf 文件（位于索引器以及向索引器发送数据的任何转发器上），确保没有任何段落将数据定向到待删除的索引。例如，如果要删除名为 "nogood" 的索引，必须确保在任何输入段落中都未出现以下属性/值对：index=nogood。删除该索引之后，索引器将放弃仍要发送到该索引的任何数据。

要删除 Splunk Web 中的索引，导航到**设置 > 索引**，然后单击您想要删除的索引右侧的**删除**。此操作将删除索引的数据目录并从 indexes.conf 中删除索引的段落。

要通过 CLI 删除索引，运行 splunk remove index 命令：

```
splunk remove index <index_name>
```

此命令将删除索引的数据目录并从 indexes.conf 中删除索引的段落。

运行索引器时，您可以运行 splunk remove index。完成命令后，您不需要重新启动索引器。

索引的删除过程通常很快，但具体时间取决于许多因素：

- 要删除的数据量。
- 您当前是否正在向同一磁盘上的其他索引中写入大量数据。
- 您要删除的索引中是否存在大量小型的 .tsidx 文件。

您还可以通过直接编辑 indexes.conf 并删除索引的段落删除索引。重新启动索引器，然后删除索引的目录。

要从索引器群集中删除索引，您必须编辑 indexes.conf 然后删除索引的段落。您不能使用 Splunk Web 或 CLI。和索引器群集上的所有类似更改一样，您首先在管理器节点上编辑文件，然后将更改应用到所有对等节点。请参阅“在索引器群集中配置对等节点索引”。在您应用 indexes.conf 更改并重新启动对等节点之后，将索引目录从各对等节点中删除。

禁用索引（而不删除）

禁用某个索引后，索引器将不再接受以此为目标的数据。但是，禁用索引不会删除索引数据，操作是可撤销的。

您可以在 Splunk Web 中禁用索引。为此，导航到**设置 > 索引**，然后单击待禁用索引右侧的**禁用**。要重新启用索引，单击索引右侧的**启用**。

您也可以使用 CLI 命令 `splunk disable index` 禁用索引：

```
splunk disable index <index_name>
```

要重新启用索引，使用 `splunk enable index` 命令。

要禁用索引器群集的索引，您必须编辑 `indexes.conf` 并在索引的段落中设置 `disabled=true`。您不能使用 Splunk Web 或 CLI。和索引器群集上的所有类似更改一样，您首先在管理器节点上编辑文件，然后将更改应用到所有对等节点。请参阅“在索引器群集中配置对等节点索引”

基于退休策略删除较旧数据

索引中的数据桶达到指定的时间或索引增长到指定的大小时，它就会滚动到“冻结”状态，此时索引器会从索引中将其删除。删除数据桶之前，索引器可以将其保存到归档中，这取决于您如何配置退休策略。

更多信息请参阅“设置退休和归档策略”。

管理用于索引并行化的管道集

索引并行化是一种允许索引器维护多个**管道**集的功能。管道集进行一系列数据处理操作，从获取原始数据、通过事件处理到将事件写入磁盘。管道集是“索引如何工作”中描述的处理管道的一个实例。因为包含各个管道，所以被称为“管道集”，例如解析管道和索引管道，它们共同构成整个处理管道

默认情况下，索引器只运行单个管道集。但是，如果底层计算机利用率很低，即同时有可用的核心和 I/O，则可以配置索引器运行两个管道集。通过运行两个管道集，您有可能使索引器的索引吞吐容量翻倍。

注意：在索引器上吞吐量的实际增加值取决于您数据导入的性质和其他因素。

此外，如果索引器难以处理突发数据，索引并行化可以帮助它容纳突发数据，如果仍然假定计算机具有可用容量的话。

总之，以下是索引并行化的一些典型用例，它们依赖于可用的计算机资源：

- 调整索引器吞吐量的规模。
- 处理突发数据。

为了更好地了解用例，并确定您的部署是否可以从多个管道集中受益，请参阅*《容量规划手册》*中的“并行化设置”。

注意：对于从 UDP 数据导入接收的指标数据，您不能对多个管道集使用索引并行化。如果您的系统使用了多个管道集，请对指标数据使用 TCP 或 HTTP 事件收集器数据导入。有关指标的更多信息，请参阅*指标*手册。

配置管道集的数量

警告：在您增加管道集的数量（默认为一个）之前，确保您的索引器可以支持多个管道集。请参阅*《容量规划手册》*中的“并行化设置”。此外，咨询专业服务，特别是如果您想要将管道集的数量增加到两个以上的话。

要将管道集的数量设为 2，请更改 `parallelIngestionPipelines` 属性（位于 `[general]` 段落，在 `server.conf` 中）：

```
parallelIngestionPipelines = 2
```

必须重新启动索引器以使更改生效。

除非有专业服务的建议，否则将管道集的最大数量限制为 2。

索引器如何处理多个管道集

当您实施两个管道集时，您就有两个完整的处理管道，从获取数据的点到将事件写入磁盘的点。管道集彼此独立运转，互相之间并不了解各自的活动。效果基本等同于每个管道集在它自己单独的索引器上运行。

索引器收到新输入后，会将该输入分配到管道集，之后管道集会通过完整的管道流程处理输入的数据，直到将输入的数据写入磁盘上的索引。

例如，如果索引器是直接引入文件，那么单个管道集会处理整个文件。管道集不共享文件的数据。类似地，当转发器开始将数据发送到索引器时，索引器会将该转发器中的整个输入分配到单个管道集。管道集会继续处理转发器中的所有数据，直到转发

器切换到另一个索引器，假设转发器跨多个索引器之间负载均衡。

各管道集可同时处理多个输入。

各管道集会写入自己的热数据桶集。

索引器如何处理将输入分配到管道集

索引器收到新输入后，例如，索引器将输入从一个刚刚与它连接的转发器分配到其中一个管道集。根据配置，索引器使用以下其中一种分配方法：

- 循环选择
- 随机加权选择

大多数情况下，首选随机加权选择。循环选择是默认方法，仅适用于运行 7.3 之前版本的索引器。

循环选择

默认情况下，索引器使用循环选择以在其管道集间分配新的输入。通过循环选择，索引器只会轮流将新输入分配到各管道集。

这个方法有个缺点，就是不考虑管道集上的当前负载。由于大小的变化和各输入的复杂性，管道集负载可能会有很大的不同。例如，通过循环选择，索引器可将下一个输入分配至重型负载管道集，同时其他管道集可保持闲置状态。

随机加权选择

随机加权选择方法考虑管道集的相对负载。索引器会随时间变化监视其管道集的加权负载，并使用该信息为数据引入选择下一个管道集，以尝试平衡相关负载。

通过随机加权方法，各管道集可向索引器报告当前处理负载的指标。索引器会根据配置时间期限查看相关负载，并在该期限内根据最新的整体负载将下一个新的输入分配到管道集。

配置选择方法

通过 `server.conf` 中的 `pipelineSetSelectionPolicy` 设置指定选择方法。

```
pipelineSetSelectionPolicy = round_robin | weighted_random
```

默认的选择方法是 `round_robin`。

您可以通过 `server.conf` 中的设置更改随机加权选择策略的行为：

- `pipelineSetWeightsUpdatePeriod`
- `pipelineSetNumTrackingPeriods`

详细信息请参阅 `server.conf` 规范文件。

查看管道集活动

您可以使用监视控制台监视部署的大多数方面。本节介绍可用来深入了解管道集性能的控制台仪表板。

监视控制台的主要文档位于*《监视 Splunk Enterprise 手册》*内。

关于管道集性能的信息，选择**索引性能：索引**菜单下的**高级**子菜单。

索引性能：“高级”仪表板基于各索引器提供管道集性能的数据。您可以使用仪表板了解管道集及其组件管道的活动。

此仪表板主要在咨询 Splunk 支持故障排除性能问题时使用。如果没有专家级的基本流程知识，很难充分解释信息决定性能问题修复。

多个管道集对于索引设置的影响

一些索引设置的作用范围是管道集。其中包括与管道、处理器或队列相关的所有设置。这些限制的示例包括在 `max_fd` 和 `maxKBps`（位于 `limits.conf` 中）以及 `maxHotBuckets`（位于 `indexes.conf` 中）。

如果您有多个管道集，则这些限制分别适用于每个管道集，而不适用于作为一个整体的索引器。例如，每个管道集分别受限于 `maxHotBuckets` 限制。如果您将 `maxHotBuckets` 设为 4，则每个管道集每次最多允许四个热数据桶，在具有两个管道集的索引器上总共允许八个热数据桶。

转发器和多个管道集

您还可以配置转发器来运行多个管道集。多个管道集增加了转发器的吞吐量，并且允许转发器同时处理多个导入。

例如，当一个转发器需要处理一个将占用管道很长一段时间的大文件时，这会特别有用。如果只有单个管道，直到转发器处理完大文件，才能处理其他文件。如果有两个管道集，当第一个管道继续处理大文件时，第二个管道可以快速地获取和转发较小的文件。

假定转发器有足够的资源，并且取决于导入数据的性质，有两个管道的转发器转发的数据可能是有一个管道的转发器转发数据的两倍。

转发器如何使用多个管道集

当您在转发器上启用多个管道集时，每个管道都会同时处理数据导入和输出。如果是在重型转发器上，每个管道也会进行分析处理。

转发器始终使用循环负载均衡在其管道集间分配新的导入。随机加权选择不可用于转发器。

转发器彼此独立地转发输出流。如果将转发器配置用于负载均衡，则它会分别对于每个输出流进行负载均衡。接收索引器会分别处理每个来自转发器的流，就好像每个流都来自不同的转发器。

注意：转发器和索引器上的管道集彼此完全独立。例如，有多个管道集的转发器可以向任意索引器转发数据，无论该索引器是有一个或两个管道集。转发器不知道索引器上的管道配置，而且它也不需要知道。同样地，有多个管道集的索引器可以从任意转发器接收数据，无论该转发器有多少个管道集。

在转发器上配置管道集

您可以使用与配置索引器相同的方式（使用 `parallelIngestionPipelines` 属性，位于 `[general]` 段落，在 `server.conf` 中）来配置转发器的管道集数量。

对于重型转发器，索引器指南适用于：底层计算机必须明显地处于低利用率。通常您应该限制管道集的数量为 2，并咨询专业服务。请参阅*《容量规划手册》*中的“并行化设置”。

对于通用转发器，单个管道集平均使用约 0.5 个核心，但利用率最大可以达到 1.5 个核心。因此，两个管道集将使用 1.0 到 3.0 个核心。如果您想要在通用转发器上配置超过两个管道集，请先咨询专业服务。

优化索引

当索引器为数据创建索引时，`splunk-optimize` 进程的一个或多个实例将间歇地运行，从而将索引文件合并在一起，以优化搜索数据时的性能。`splunk-optimize` 进程会短时间地占用大量 CPU。

如果没有经常运行 `splunk-optimize`，搜索效率就会下降。

`splunk-optimize` 仅在热**数据桶**上运行。如果您发现温数据桶中包含更多索引文件（`.tsidx`），则可手动在温数据桶上运行该进程；通常，文件数量为超过 25 个。要运行 `splunk-optimize`，请转到 `$SPLUNKHOME/bin` 并键入：

```
splunk-optimize -d|--directory <bucket_directory>
```

`splunk-optimize` 接受许多可选参数。要查看可用参数的列表，键入：

```
splunk-optimize
```

要开启从 `splunk-optimize` 到 `splunkd.log` 的 verbose 登录，您可以将 `log.cfg` 中的 `category.SplunkOptimize` 设为 INFO 或 DEBUG。这样做的推荐做法是通过 CLI：

```
splunk set log-level SplunkOptimize -level DEBUG -auth admin:passwd
```

更多关于数据桶的信息，请参阅“Splunk 如何存储索引”。

使用监视控制台查看索引性能

您可以使用监视控制台监视部署的大多数方面。本主题介绍可用来深入了解索引性能的控制台仪表板。

监视控制台的主要文档位于*《监视 Splunk Enterprise 手册》*内。

在**索引**菜单下方有两个性能仪表板：

- 索引性能：实例
- 索引性能：部署

正如它们的名字所示，它们会提供类似的信息，范围限于单个实例或是整个部署。

性能仪表板提供索引创建过程中的各种信息，例如：

- 索引速度
- 队列填充率
- 数据管道各个部分的 CPU 信息

有关更多信息，请查看仪表板本身。此外，请参阅*《监视 Splunk Enterprise》* 中的“索引性能仪表板”。

管理索引存储

索引器如何存储索引

索引器为您的数据创建索引时，会创建许多文件：

- 压缩形式的原始数据（**原始数据日志**）
- 指向原始数据的索引（**tsidx 文件**）
- 其他一些元数据文件

这些文件共同组成了 Splunk Enterprise **索引**。这些文件驻留在按时间组织的目录集或**数据桶**中。各数据桶包含原始数据日志文件以及相关的 tsidx 和元数据文件。各数据桶中的数据受有限的时间范围约束。

索引通常由很多数据桶组成，数据桶数量增加，索引也会相应变大。当数据继续进入系统时，索引器会创建新的数据桶容纳增加的数据。

索引器上的部分数据桶包含新建索引的数据；其他目录包含以前索引的数据。索引中的数据桶数量会变得非常大，具体取决于您正在索引的数据量以及您保留数据的时长。

具体如何，一切取决于您。

默认情况下，索引器按照通过多个阶段使数据逐渐衰退的方式来处理已索引的数据。在经过一长段时间（通常为几年）后，索引器将从系统中删除旧数据。Splunk 使用的默认方案应该会满足您的需求。

不过，如果您要为大量数据创建索引、拥有特定数据保存要求或者需要仔细规划老化策略，则需要阅读本主题。另外，如果要备份数据，本主题有助于您了解数据的位置。因此，请继续阅读....

数据如何老化

数据桶在老化时会经历多个阶段：

- 热
- 温
- 冷
- 冻结
- 解冻

当数据桶老化时，它们从一个阶段“滚动”到下一个阶段。首次索引数据时，会将数据写入热数据桶。热数据桶是可主动向其中写入内容的数据桶。一个索引每次可以打开若干个热数据桶。热数据桶也是可搜索的。

当满足一定的条件（例如，热数据桶达到特定大小或者重新启动索引器）时，热数据桶将成为温数据桶（“滚动到温数据桶”），并将在其位置创建一个新的热数据桶。温数据桶是重新命名的，但是保留在还是热数据桶时所在的相同位置。温数据桶是可搜索的，但不能主动向其中写入内容。温数据桶数量可以很大。

一旦满足进一步的条件（例如，索引达到温数据桶的某些最大值）时，索引器会开始基于时间将温数据桶滚动到冷数据桶。它会始终选择将时间最长的温数据桶滚动到冷数据桶。当数据桶老化时，它们将继续以此方式滚动到冷数据桶。冷数据桶驻留位置和热数据桶和温数据桶不同。您可以配置位置，这样冷数据桶会驻留在更廉价的存储上。

最后，当某些特定的其他基于时间或基于大小的条件满足后，冷数据桶会滚动到冻结状态，这时，在有选择地进行归档之后，会将这些数据桶从索引中删除。

如果冻结的数据已归档，则稍后它可以被解冻。解冻数据桶中的数据是可搜索的。

Indexes.conf 中的设置决定了何时将数据桶从一种状态移动到另一种状态。

下面是数据桶老化时会经过的几个阶段：

数据桶阶段	描述	是否可搜索？
热	新数据写入热数据桶。每个索引都有一个或多个热数据桶。	是
温	从热数据桶滚动来的数据桶。不能将新数据写入温数据桶。索引可以有很多温数据桶。	是
冷	从温数据桶滚动而来并移动到其他位置的数据桶。索引可以有很多冷数据桶。	是

冻结	从冷数据桶滚动来的数据桶。索引器会删除冻结的数据桶，但您可以选择首先将其归档。归档的数据桶稍后可解冻。	否
解冻	从存档中恢复数据桶。如果您将冻结的数据桶归档，则稍后可以通过将其解冻使其返回到索引中。	是

注意：SmartStore 功能会将数据放入 S3 这样的远程存储中，因此，对于使用 SmartStore 功能启用的索引，一般来说不存在冷阶段。请参阅“数据桶阶段和 SmartStore”。

索引目录的结构

每个索引都在 `$SPLUNK_HOME/var/lib/splunk` 下占用自己的目录。目录名称与索引名称相同。索引目录下是按状态（热/温、冷或解冻）对数据桶进行分类的一系列子目录。

每个数据桶就是这些目录的子目录。数据桶名称表明了他们包含的数据时间。

下面是默认索引 (`defaultdb`) 的目录结构：

数据桶阶段	默认位置	注释
热	`$SPLUNK_HOME/var/lib/splunk/defaultdb/db/*`	每个热数据桶占用自己的子目录。
温	`$SPLUNK_HOME/var/lib/splunk/defaultdb/db/*`	每个温数据桶占用自己的子目录。
冷	`$SPLUNK_HOME/var/lib/splunk/defaultdb/colddb/*`	每个冷数据桶占用自己的子目录。当温数据桶滚动到冷数据桶时，它们会移动到此目录中。
冻结	数据桶冻结时，这些数据桶会被删除或归档到您指定的位置。	默认是删除。有关如何改为归档数据的信息，请参阅“归档索引的数据”。
解冻	`$SPLUNK_HOME/var/lib/splunk/defaultdb/thaweddb/*`	被归档或之后解冻的数据桶会留在此目录中。有关将归档数据恢复到解冻阶段的信息，请参阅“恢复归档的数据”。

热数据桶、温数据桶、冷数据桶的路径和解冻目录是可配置的。请参阅“配置索引存储”和“对索引数据使用多个分区”。

所有索引位置必须是可写位置。

注意：在 6.0 之前版本的 Splunk Enterprise 中，复制的索引器群集数据桶副本始终驻留在 `colddb` 目录，即使它们是热或温数据桶。自 6.0 版本起，热和温复制的副本驻留在 `db` 目录中，非复制副本也相同。

数据桶命名约定

数据桶的名称取决于：

- 数据桶的阶段：热或温/冷/解冻
- 数据桶目录的类型：非群集、群集生成或群集复制

重要提示：数据桶命名约定易受变化影响。

非群集数据桶

独立索引器会创建非群集数据桶。这些数据桶使用一种命名约定。

群集数据桶

作为**索引器群集**一部分的索引器会创建群集数据桶。群集数据桶有多个精确的副本。群集数据桶的命名约定区分了副本的类型，原始或复制。

简而言之，根据其**复制因子**，索引器群集中的数据桶会有多个副本。当数据进入群集时，接收索引器将数据写入热数据桶。该接收索引器被称为数据来源群集**节点**，而数据写入的数据桶被称为数据桶的原始副本。

当数据写入热副本时，数据来源对等节点将热数据以数据块的形式流送到群集中其他索引器上。这些索引器被称为数据桶的目标对等节点。目标对等节点上流数据的副本被称为数据桶的复制副本。

当数据来源对等节点将其原始热数据桶滚动为温数据桶时，目标对等节点会滚动该数据桶的复制副本。温副本是彼此的精确副本。

有关索引器群集架构以及复制的数据流的介绍，请参阅“基本索引器群集架构”。

数据桶名称

以下是命名约定：

数据桶类型	热数据桶	温/冷/解冻数据桶
非群集	hot_v1_<localid>	db_<newest_time>_<oldest_time>_<localid>
群集生成	hot_v1_<localid>	db_<newest_time>_<oldest_time>_<localid>_<guid>
群集复制	<localid>_<guid>	rb_<newest_time>_<oldest_time>_<localid>_<guid>

注意：

- <newest_time> 和 <oldest_time> 是表示数据桶中数据时间的时间戳。时间戳以 UTC epoch 时间（秒）表示。例如：db_1223658000_1223654401_2835 是一个包含从 2008 年 10 月 10 日 16 点到 17 点数据的非群集温数据桶。
- <localid> 是数据桶的 ID。对于群集数据桶，数据桶的原始和复制副本具有相同的 <localid>。
- <guid> 是数据来源对等节点的 guid。guid 位于对等节点的 $SPLUNK_HOME/etc/instance.cfg 文件中。

在索引器群集中，原始温数据桶及其复制副本具有相同的名称，只是前缀不同（原始数据桶的前缀为 db，复制副本的前缀为 rb）。

注意：在索引器群集中，数据从数据来源对等节点流送到目标对等节点时，数据首先会进入目标对等节点上的临时目录（由 <localid>_<guid> 的热数据桶约定进行标识）。这适用于所有复制数据桶副本，无论流送的数据桶是否为热数据桶。例如，在数据桶修复活动期间，一个对等节点可能会将温数据桶流送到其他对等节点。当该数据桶的复制完成时，<localid>_<guid> 目录会滚动为温数据桶目录（由 rb_ 前缀进行标识）。

数据桶和 Splunk Enterprise 管理

管理 Splunk Enterprise 时，本节内容可帮助您了解索引器如何在数据桶中存储索引。特别是，有多个管理活动需要对数据桶有更好的理解：

- 有关设置退休和归档策略的信息，请参阅“设置退休和归档策略”。退休策略可以基于数据的大小，也可以基于数据的时间。
- 有关如何归档索引数据的信息，请参阅“归档索引的数据”。要了解如何恢复归档数据，请参阅“恢复归档的数据”。
- 要了解如何备份数据，请参阅“备份索引数据”。该主题中还介绍了如何手动将热数据桶滚动到温数据桶，以便随后可以对其进行备份。
- 有关设置磁盘使用限制的信息，请参阅“对磁盘使用情况设置限制”。
- 有关可配置的数据桶设置的列表，请参阅“配置索引存储”。
- 有关配置索引大小的信息，请参阅“配置索引大小”。
- 有关对索引数据进行分区的信息，请参阅“对索引数据使用多个分区”。
- 有关在索引器群集中数据桶如何工作的信息，请参阅“数据桶和索引器群集”。
- 有关数据桶和 SmartStore 的信息，请参阅“数据桶阶段和 SmartStore”。

此外，您还可以参阅*《管理员手册》*中的 "indexes.conf"。

配置索引存储

您在 indexes.conf 中对索引进行配置。您对 indexes.conf 的编辑方式取决于您是否使用了**索引复制**，也称为索引器群集化：

- **对于非群集索引，**编辑 $SPLUNK_HOME/etc/system/local/ 中的 indexes.conf 版本；如果不存在，可创建一个配置文件。不要编辑 $SPLUNK_HOME/etc/system/default 中的副本。有关配置文件和目录位置的信息，请参阅“关于配置文件”。
- **对于群集索引**，在群集管理器节点上创建或编辑 indexes.conf 版本，然后将其分发到所有对等节点上，如“在索引器群集中配置对等节点索引”中所述。

仅对于非群集索引，可以选择使用 Splunk Web 来配置索引的路径。转到**设置 > 服务器设置 > 常规设置**。在**索引设置**部分下，设置字段**索引路径**。完成此操作后，必须从 CLI（而不是从 Splunk Web 中）重新启动索引器。然而，大多数其他设置需要直接编辑 indexes.conf。

影响索引数据桶的属性

下表列出了影响**数据桶**的关键 indexes.conf 属性以及其配置的内容。该表中还提供了有关如何使用这些属性的其他主题的链接。有关这些属性及其他内容的最详细信息，始终可以参阅 indexes.conf 规范文件。

注意：此列表针对非 SmartStore 索引。控制 **SmartStore** 索引的一组属性有所不同。请参阅“配置 SmartStore”。

属性	配置内容	默认	有关更多信息，请参阅……
homePath	包含热数据桶和温数据桶的路径。（**必需。**） 该位置必须是可写位置。	$SPLUNK_HOME/var/lib/splunk/defaultdb/db/ （仅适用于默认索引）	配置索引路径属性
coldPath	包含冷数据桶的路径。（**必需。**） 该位置必须是可写位置。	$SPLUNK_HOME/var/lib/splunk/defaultdb/colddb/ （仅适用于默认索引）	配置索引路径属性
thawedPath	包含所有解冻数据桶的路径。（**必需。**） 该位置必须是可写位置。	$SPLUNK_HOME/var/lib/splunk/defaultdb/thaweddb/ （仅适用于默认索引）	配置索引路径属性
repFactor	确定是否将索引复制到其他群集对等节点。（**对于群集对等节点上的索引，这是必需的。**）	0（表示不将索引复制到其他对等节点；这是非群集索引的正确行为）。对于群集索引，您*必须*将 repFactor 设置为 auto，使索引复制到其他对等节点。	在索引器群集中配置对等节点索引
maxHotBuckets	并发热数据桶的最大数量。该值应至少为 2，以便处理任何归档数据。例如，**main** 默认索引将此值设置为 10。	3（对于新的自定义索引）。	数据如何老化
maxDataSize	**确定热到温的滚动行为。**热数据桶的最大大小。如果热数据桶达到此大小，将滚动到温数据桶。此属性还用于确定所有数据桶的近似大小。	特定值 "auto"，将大小设为 750MB。	数据如何老化
maxWarmDBCount	**确定温到冷的滚动行为。**温数据桶的最大值。如果达到最大值，温数据桶开始滚动到冷数据桶。	300	对索引数据使用多个分区
maxTotalDataSizeMB	**确定冷到冻结的滚动行为。**索引的最大大小。如果达到此限制，冷数据桶会开始滚动到冻结数据桶。	500000（MB）	设置退休和归档策略
frozenTimePeriodInSecs	**确定冷到冻结的滚动行为。**某个数据桶的最长时间，之后，它将滚动到冻结数据桶。	188697600（秒；大约 6 年）	设置退休和归档策略
coldToFrozenDir	归档数据的位置。确定数据桶从冷滚动到冻结的行为。如果设置此属性，索引器会在从索引中删除冻结数据桶之前，将冻结数据桶归档到此目录。	如果未设置此属性或 coldToFrozenScript，索引器将只是记录该数据桶的目录名称，然后在该数据桶滚动到冻结之后立即将其删除。	归档索引的数据
coldToFrozenScript	在冷数据桶滚动到冻结数据桶之前要运行的脚本。如果同时设置此属性和 coldToFrozenDir，索引器将使用 coldToFrozenDir 并忽略此属性。	如果未设置此属性或 coldToFrozenDir，索引器将只是记录该数据桶的目录名称，然后在该数据桶滚动到冻结之后立即将其删除。	归档索引的数据
homePath.maxDataSizeMB **coldPath.maxDataSizeMB**	homePath（热/温数据桶存储）或 coldPath（冷数据桶存储）的最大大小。如果其中任一属性缺失或设置为 0，其路径在大小方面不会单独受到约束。	无	按照数据桶类型配置索引大小
maxVolumeDataSizeMB	卷的最大大小。如果未设置此属性，则个别卷的大小将不受限制。	无	使用卷配置索引大小

配置索引路径属性

创建新索引时，您可以配置若干个索引路径属性，例如 homePath 和 coldPath。配置路径属性时，请遵循以下限制和建议：

- 该路径必须是可写路径。如果是 homePath，则父路径也必须是可写路径。
- 请勿在索引路径中使用环境变量。唯一一个特例是 SPLUNK_DB。
- 路径不能为根目录，例如 homePath=/myindex 或 homePath=C:\myindex。
- 建议您使用 $_index_name 作为索引名称的占位符来指定路径。例如：

```
homePath = $SPLUNK_DB/$_index_name/db
```

在运行时，索引器将 $_index_name 扩展为索引的名称。例如，若索引名称为 "newindex"，则 homePath 变成 $SPLUNK_DB/newindex/db。

索引路径属性集包括：

- homePath
- coldPath
- thawedPath
- bloomHomePath
- summaryHomePath
- tstatsHomePath

有关路径属性的更多信息，请参阅 indexes.conf 规范文件。

有关使用多个分区来存储索引数据的信息，请参阅“对索引数据使用多个分区”。

索引大小和索引器群集

注意：这部分仅适用于非 SmartStore 索引。群集会以不同的方式处理 SmartStore 索引大小。请参阅“为 SmartStore 索引配置数据保留”。

控制非 SmartStore 索引大小的属性和数据桶的数量独立作用于每一个对等节点。他们不会作用于整个群集。

例如，以 maxTotalDataSizeMB 属性为例。该属性指定索引的最大大小。属性值应用于每个对等节点，以限制每个对等节点的索引大小。当特定对等节点的索引达到其最大大小时，此对等节点会冻结其索引副本中最旧的数据桶。

这意味着，对等节点的索引大小取决于节点上该索引的所有数据桶副本的总大小。这些副本是主要副本、可搜索副本、不可搜索副本还是过多副本并不重要。所有这些副本的大小都会纳入对等节点索引大小的计算中。

由于群集通知不会非常平均的将数据桶副本分发给一组对等节点，所以一个索引在不同对等节点的大小通常不同。这表示，该索引可能在某个对等节点上已达其最大大小，但在其他对等节点上却仍有成长空间。

为了解决这个问题，每个对等节点在冻结数据桶的副本时会通知管理器节点。从此时起，管理器节点就不会再为冻结的数据桶发起任何修复活动。但是，管理器节点不会指示其他对等节点冻结该数据桶的副本。所以，每个对等节点只会在该索引的副本达到最大大小限制时冻结该数据桶的副本。请参阅“群集如何处理冻结数据桶”。

注意：尽管这些属性单独作用于每个对等节点，仍应将群集内所有对等节点的属性设为相同值。请参阅“在索引器群集中配置对等节点索引”。

有关调整群集磁盘空间需求的帮助信息，请参阅“存储注意事项”。

原始数据日志压缩

当索引器为数据建立索引时，它会将数据写入到**原始数据日志**中。索引器在此过程中会对数据进行压缩。默认情况下，压缩使用 gzip 算法，但是您可以通过 indexs.conf 中的 journalCompression 设置更改压缩算法。可用的压缩算法包括：

- gzip
- lz4
- zstd

如果更改了压缩算法，则会使用新方法来压缩新数据桶中的日志，但是现有数据桶中的日志将继续通过最初为其建立索引时所用的压缩方法进行存储。

移动索引数据库

可将索引数据库从一个位置移动到另一个位置。您可以通过个人操作系统的命令行界面修改 SPLUNK_DB 的路径定义来实现这一操作。

本主题中的步骤会假定索引数据库位于默认位置，而且是在安装期间创建的。

如果将单个索引或索引的一部分移动到不同位置，则此主题中的步骤无效。有关 Splunk Enterprise 索引结构的信息，请参

阅“索引器如何存储索引”。有关如何更改单个索引位置的信息，请参阅“配置索引存储”。

注意：尽管您可以使用 Splunk Web 更改单个索引的位置或索引量，但无法用它来更改索引的默认存储位置，SPLUNK_DB。

对于 *nix 用户

先决条件

确保目标文件系统拥有的空间至少是您计划创建索引的原始数据总大小的 1.2 倍。

步骤

1. 创建目标目录，并且 Splunk Enterprise 运行时所使用的用户身份对其具有写入权限。例如，如果 Splunk Enterprise 以用户 "splunk" 的身份运行，为其指定该目录的所有权：

```
mkdir /foo/bar
chown splunk /foo/bar/
```

有关设置 Splunk Enterprise 运行时所使用用户的信息，请参阅*《安装手册》*中的“以其他或非 root 用户身份运行 Splunk Enterprise”。

2. 停止索引器：

```
splunk stop
```

3. 将索引文件系统复制到目标目录：

```
cp -rp $SPLUNK_DB/* /foo/bar/
```

4. 取消设置 SPLUNK_DB 环境变量：

```
unset SPLUNK_DB
```

5. 更改 SPLUNK_DB 属性（位于 $SPLUNK_HOME/etc/splunk-launch.conf 中）来指定新索引目录：

```
SPLUNK_DB=/foo/bar
```

6. 启动索引器：

```
splunk start
```

索引器将选取它停止索引新副本、从中读取内容和向其中写入内容的位置。

7. 在确认索引器能够读取和写入新位置后，可删除旧的索引数据库。

对于 Windows 用户

先决条件

确保目标驱动器或目录拥有的空间至少是您计划创建索引的原始数据总大小的 1.2 倍。

警告：请勿将映射的网络驱动器用于索引存储。

步骤

1. 在命令提示符下，确保目标目录具有允许 splunkd 进程向该目录写入的权限：

```
C:\Program Files\Splunk> D:
D:\> mkdir \new\path\for\index
D:\> cacls D:\new\path\for\index /T /E /G <the user Splunk Enterprise runs as>:F
```

有关确定 Splunk Enterprise 运行时所使用用户的详细信息，请参阅*《安装手册》*中的“在 Windows 上安装”。

2. 停止索引器：

```
splunk stop
```

您也可使用服务控制面板来停止 splunkd 和 splunkweb 服务。

3. 将现有索引文件系统复制到目标目录：

```
xcopy "C:\Program Files\Splunk\var\lib\splunk\*.*" D:\new\path\for\index /s /e /v /o /k
```

4. 取消设置 SPLUNK_DB 环境变量：

```
set SPLUNK_DB=
```

5. 编辑 SPLUNK_DB 属性（位于 %SPLUNK_HOME%\etc\splunk-launch.conf 中）来指定新索引目录：

```
SPLUNK_DB=D:\new\path\for\index
```

如果配置文件中包含 SPLUNK_DB 属性的行将英镑符号 (#) 作为其第一个字符，则删除 #。

6. 启动索引器：

```
splunk start
```

索引器将选取它停止索引新副本、从中读取内容和向其中写入内容的位置。

7. 在确认索引器能够读取和写入新位置后，可删除旧的索引数据库。

对索引数据使用多个分区

注意：本主题与 **SmartStore** 索引无关。请参阅“关于 SmartStore”。

索引器可对其索引数据使用多个分区。可以将索引器配置为基于多个索引和**数据桶**类型使用多个磁盘/分区/文件系统，但前提是正确安装这些磁盘/分区/文件系统，并通过 indexes.conf 正确地指向它们。但是，在大多数情况下，最佳做法是使用单个高性能文件系统来存储索引数据。

如果确实使用了多个分区，安排索引数据的最常用方式是在本地计算机上保存热/温数据桶，而将冷数据桶放置在单独的磁盘阵列上长期存储。您需要在具有快读取/写入分区的计算机上存储热/温数据桶，因为，大多数搜索将在其中进行。

配置多个分区

配置多个分区：

1. 按照您通常在任意操作系统中设置分区的方式来设置分区。

2. 安装磁盘/分区。

3. 编辑 indexes.conf 以指向分区的正确路径。您基于“按索引”设置路径，因此，可为不同的索引设置单独的分区。每个索引有自己的 [<index>] 段落，其中 <index> 是索引的名称。下面是主要路径设置：

- homePath 是包含索引的热数据桶和温数据桶的路径。
- coldPath 是包含索引的冷数据桶的路径。
- thawedPath 是包含索引的任何解冻数据桶的路径。

有关定义索引路径的指导原则，请参阅“配置索引路径属性”。

配置最大索引大小

注意：本主题与 **SmartStore** 索引无关。请参阅“为 SmartStore 索引配置数据保留”。

可以通过多种方式配置最大索引大小：

- 基于“按索引”
- 分别针对热/温和冷**数据桶**

- 跨多个索引，使用卷

要配置索引存储大小，需在 indexes.conf 中设置属性。有关本主题中所提及属性的详细信息，请参阅“配置索引存储”。

警告：处理索引时，索引器可能偶尔会在短时间内超出所配置的最大空间。在设置限制时，请务必考虑一些缓冲区空间。另请注意，特定的系统（如大多数 Unix 系统）将在它们的分区上保留可配置的预留空间。如果具有这样的预留空间，则在确定您的索引会增长的多寡时必须考虑该预留空间。

配置每个索引的索引大小

要基于“按索引”设置最大索引大小，请使用 maxTotalDataSizeMB 属性。如果达到此限制，数据桶开始滚动到冻结数据桶。

按照数据桶类型配置索引大小

要设置 homePath（热/温数据桶存储）或 coldPath（冷数据桶存储）的最大大小，请使用 maxDataSizeMB 设置：

```
# set hot/warm storage to 10,000MB
homePath.maxDataSizeMB = 10000
# set cold storage to 5,000MB
coldPath.maxDataSizeMB = 5000
```

可全局设置 maxDataSizeMB 属性，也可单独为每个索引设置。索引级别设置将覆盖全局设置。要在多组索引中控制数据桶存储，请使用下文所述的 maxVolumeDataSizeMB 属性。

当 homePath 目录大小超出 homePath.maxDataSizeMB 时，索引器会将最旧的温数据桶滚动为冷数据桶，将数据桶移动到 coldPath 目录下。

当 coldPath 目录大小超出 coldPath.maxDataSizeMB 时，索引器会将最旧的冷数据桶滚动为冻结数据桶。

使用卷配置索引大小

可跨多个索引管理磁盘使用情况，方法是创建卷并为卷指定最大数据大小。一个卷代表文件系统上一个驻留已创建索引数据的目录。

卷可存储来自多个索引的数据。通常，对热/温数据桶和冷数据桶使用单独的卷。例如，可设置一个卷来包含所有索引的热/温数据桶，设置另外一个卷来包含冷数据桶。

您可以使用卷来定义 homePath 和 coldPath。但不能使用卷来定义 thawedPath。

此外，如果显式定义了 bloomHomePath，则必须使用卷。

配置卷

要设置卷，请使用以下语法：

```
[volume:<volume_name>]
path = <pathname_for_volume>
```

也可选择包括一个 maxVolumeDataSizeMB 属性，用于指定该卷的最大大小。

例如：

```
[volume:hot1]
path = /mnt/fast_disk
maxVolumeDataSizeMB = 100000
```

该示例定义了一个名为 "hot1" 的卷，该卷位于 /mnt/fast_disk，最大大小为 100,000MB。

类似地，此段落定义一个名为 "cold1" 的卷，最大大小为 150,000MB：

```
[volume:cold1]
path = /mnt/big_disk
maxVolumeDataSizeMB = 150000
```

使用卷

卷配置好后，您即可使用这些卷来定义索引的 homePath 和 coldPath。例如，您可以使用上面配置好的卷定义两个索引：

```
[idx1]
homePath = volume:hot1/idx1
coldPath = volume:cold1/idx1

[idx2]
homePath = volume:hot1/idx2
coldPath = volume:cold1/idx2
```

您可使用您觉得合理的方式来使用卷以管理索引存储空间。但通常，由于存在不同的存储要求（特别是在处理不同的数据桶类型时），卷与热/温数据桶和冷数据桶相关联。因此，您可能会将某些卷专门用于指定 homePath（热/温数据桶），将另一些卷用于 coldPath（冷数据桶）。

当包含温数据桶的卷达到其 maxVolumeDataSizeMB 时，开始将数据桶滚动到冷数据桶。当包含冷数据桶的卷达到其 maxVolumeDataSizeMB 时，开始将数据桶滚动到冻结数据桶。如果某个卷同时包含温数据桶和冷数据桶（如果某个索引的 homePath 和 coldPath 同时设置为同一个卷，就会出现这种情况），则时间最长的数据桶将滚动到冻结数据桶。

组合在一起

以下示例显示如何将每个索引的 homePath.maxDataSizeMB 和 coldPath.maxDataSizeMB 属性与卷结合使用，以保持对索引存储精细粒度的控制。该示例尤其显示了如何使用这些属性来防止一个索引内的突发数据诱发其他索引的大规模数据桶移动。您可以使用每个索引的设置来确保索引占用的空间不会超过指定的大小，从而缓解大规模数据桶移动的问题。

```
# global settings

# Inheritable by all indexes: No hot/warm directory (homePath) can exceed 1 TB.
# Individual indexes can override this setting.
homePath.maxDataSizeMB = 1000000

# volumes

[volume:caliente]
path = /mnt/fast_disk
maxVolumeDataSizeMB = 100000

[volume:frio]
path = /mnt/big_disk
maxVolumeDataSizeMB = 1000000

# indexes

[i1]
homePath = volume:caliente/i1
# homePath.maxDataSizeMB is inherited from the global setting
coldPath = volume:frio/i1
# coldPath.maxDataSizeMB not specified anywhere:
# This results in no size limit - old-style behavior

[i2]
homePath = volume:caliente/i2
homePath.maxDataSizeMB = 1000
# overrides the global default
coldPath = volume:frio/i2
coldPath.maxDataSizeMB = 10000
# limits the size of cold buckets

[i3]
homePath = /old/style/path
homePath.maxDataSizeMB = 1000
coldPath = volume:frio/i3
coldPath.maxDataSizeMB = 10000
```

对磁盘使用情况设置限制

注意：本主题与 **SmartStore** 索引无关。请参阅“根据缓存磁盘分区的占用情况启动逐出”，以获取有关 SmartStore 如何控制本地磁盘使用情况的信息。

Splunk Enterprise 通过多种方法来控制磁盘空间。索引将消耗大部分磁盘空间。如果磁盘空间不足，索引器将停止创建索引。您可设置一个最小可用空间限制，来控制可用磁盘空间减少到多少，才会停止创建索引。一旦空间超出最小值，将恢复索引创建。

注意：要确定索引所需的空间大小，请参阅《*容量规划手册*》中的“评估存储要求”。

设置最小可用磁盘空间

您可为存储索引数据的磁盘设置最小可用磁盘的空间。如果达到该限制，索引器将停止操作。索引创建和搜索都会受到影响：

- 索引器会定期检查所有包含索引的分区上的空间。如果其中任何分区已达到可用磁盘空间限制，索引器将停止为数据创建索引，直到有更多空间可用。系统将发送 UI 横幅和 splunkd 警告，表示需要清理更多磁盘空间。
- 在尝试启动搜索之前，索引器将要求存储 dispatch 目录的文件系统上有指定大小的可用空间，$SPLUNK_HOME/var/run/splunk/dispatch

默认最小可用磁盘空间为 5000MB。

注意：

- 通过这种方法，索引器并不清理任何磁盘空间。它只是暂停操作，直到有更多空间可用。
- 暂停索引创建时，可能会丢失传入数据。

通过 Splunk Web、CLI 或 server.conf 配置文件，可设置最小可用磁盘空间。

在 Splunk Web 中

在 Splunk Web 中指定最小磁盘使用：

1. 单击 Splunk Web 右上角的**设置**。
2. 单击**服务器设置**。
3. 单击**常规设置**。
4. 在**索引设置**部分下，将字段**可用磁盘空间（MB）不足时暂停索引**设为所需的最小可用磁盘空间（MB）。
5. 单击**保存**。
6. 重新启动索引器，使更改生效。

通过命令行界面（CLI）

可使用 CLI 设置最小可用磁盘空间。以下示例将最小可用磁盘空间设置为 20,000MB（20GB）：

```
splunk set minfreemb 20000
splunk restart
```

有关使用 CLI 的信息，请参阅《管理员手册》中的“关于 CLI”。

在 server.conf 中

您还可在 server.conf 文件中设置最小可用磁盘空间。相关段落/属性如下所示：

```
[diskUsage]
minFreeSpace = <num>
```

请注意，<num> 代表兆字节。默认值为 5000。

控制索引存储

indexes.conf 文件包含索引配置设置。通过指定最大索引大小或数据的最长时间可控制磁盘存储使用情况。如果达到其中的一个限制，将删除（默认操作）或归档时间最长的索引数据。通过使用预定义归档脚本或创建您自己的归档脚本，可对数据归档。

有关如何使用 indexes.conf 设置最大索引大小或时间的详细说明，请参阅“设置退休和归档策略”。

有关索引存储的详细信息，请参阅“索引器如何存储索引”。

减少 tsidx 磁盘使用量

有 2 个选项可用于最小化 **tsidx 文件**使用的磁盘空间。您可以通过使用 tsidxWritingLevel 来配置其他压缩和优化，并使用 tsidx 保存策略来计划删除 tsidx 文件的时间。

tsidx 保存策略

tsidx 保存策略决定索引器保留 tsidx 文件的时长；索引器使用这些文件在数据内进行快速、高效的搜索。默认情况下，索引器只要保留了索引数据本身，就会一直保留这些数据的 tsidx 文件。您可以调整策略将旧数据的 tsidx 文件删除，从而实现存储成本与搜索性能之间的最佳平衡。

索引器将 tsidx 文件与 **rawdata 文件**一起存储在**数据桶**中。tsidx 文件对于实现大量数据范围内的有效搜索非常重要。但它们也占用了大量的存储空间。

对于日常要定期进行搜索的数据，则绝对需要 tsidx 文件。但是，如果有些数据在很长的一段时间内只会偶尔进行搜索，则可以调整 tsidx 保存策略，以便在 tsidx 文件达到一定时间时即予以减少。这种做法可减少索引数据所占用的磁盘空间。

Tsidx 减少进程将删除完整的 tsidx 文件，并把它们替换为包含基本元数据的迷你版 tsidx 文件。原始数据文件和其他元数据文件不会受影响。如果需要，您仍可以在过期的数据中执行搜索，但这类搜索的性能将显著恶化。特别是罕见术语搜索，这种搜索的运行会很慢。

总而言之，tsidx 减少的主要使用案例是多数搜索都是基于近期数据的场景。在此场景中，对旧数据的快速访问可能不值得在存储相应 tsidx 文件上投入的成本。若是减少旧数据的 tsidx 文件，这对于大多数搜索的性能不会产生什么影响，但却节约了大量的磁盘存储空间。

评估节省的存储空间

Tsidx 减少实际上是用文件的较小版本（称为迷你版 tsidx 文件）来替代数据桶的最大大小的 tsidx 文件。同时消除了数据桶的 merged_lexicon.lex 文件。

最大大小的 tsidx 文件通常占据了数据桶整个空间的一大部分。具体的量取决于数据类型。包含许多唯一术语的数据要求更大的 tsidx 文件。通用指导原则是，tsidx 减少进程将数据桶大小减少约 1/3 到 2/3。例如，可将 1GB 的数据桶大小减少到 350MB 到 700MB 之间。

若要粗略评估一下数据桶的可减少空间，可查看其 merged_lexicon.lex 文件的大小。merged_lexicon.lex 文件是说明数据桶数据中唯一术语数量的指示器。merged_lexicon.lex 文件较大的数据桶则其 tsidx 文件可以减少的程度更大，因为其中有大量的唯一术语。

迷你版 tsidx 文件的大小一般约为其对应原始文件最大大小的 5% 到 10%。但是正如前面所提到的，数据桶大小的总体减少量没有这么多 - 通常为 1/3 到 2/3 之间。这是因为除迷你版 tsidx 文件外，**减少的数据桶**还保留了原始数据文件和一些元数据文件。

tsidx 减少的工作原理

在启用 tsidx 减少后，可以基于每个索引指定减少时间。当索引数据桶达到指定时间时，索引器会将其 tsidx 文件减少。

减少进程

默认情况下，tsidx 减少进程每十分钟运行一次。该进程检查索引中的每个数据桶。若有数据桶最近的事件符合指定的减少时间，则此进程会开始减少 tsidx 文件。

减少进程每次只能运行于一个数据桶。如果有多个数据桶待进行大小减少，则此进程会依次处理。

减少进程所需时间很短。例如，当运行于 1GB 数据桶时，减少进程通常只需要几秒钟即可完成。

tsidx 文件减少后就会保持在减少后的状态。如果将 tsidx 减少设置为禁用或增加指定的减少时间，这一变更只会影响大小尚未减少的数据桶。但是，在必要时还是有一种方式，可以将大小减少的数据桶还原成带完整 tsidx 文件的数据桶。请参阅“将大小减少的数据桶恢复到原始状态”。

数据桶文件减少后的影响

tsidx 减少进程将每个目标数据桶中的最大大小 tsidx 文件替换为包含基本元数据的迷你版文件，从而实现完整文件的消除。迷你版 tsidx 文件由原始 tsidx 文件标头组成，包含每个事件的元数据。另外，tsidx 减少也消除了数据桶的

merged_lexicon.lex 文件。

数据桶保留了其原始数据文件、迷你版 tsidx 文件和部分其他元数据文件，包括 bloomfilter 文件。

最大大小的 tsidx 文件的扩展名为 .tsidx。迷你版 tsidx 文件的扩展名为 .mini.tsidx。

最大大小的 tsidx 文件只有在迷你版文件创建后才会删除。这表示，在很短的一段时间内，数据桶会同时包含两个版本的文件，从而增加了磁盘空间的使用量。

减少对正在执行的搜索的影响

如果符合进行 tsidx 减少要求的特定数据桶正在执行搜索，该数据桶大小的减少会延至此搜索完成后再进行。迷你版 tsidx 文件会及时创建，但完整文件的删除会等到搜索完成后再进行。

如果索引器正执行一个跨多个数据桶的搜索，其中包括一个待进行大小减少的数据桶，则此数据桶的大小减少可能在此搜索到达该数据桶前就会完成。如无意外，当搜索到达大小减少的数据桶后，此搜索在此数据桶的运行会很慢。

跨大小减少后数据桶的搜索

一旦数据桶完成了 tsidx 减少，您即可在此数据桶运行搜索，但这些搜索需要更长的时间才能完成。由于索引器优先搜索最新的数据桶，所以会在到达大小减少后数据桶之前即返回其他大小未减少数据桶的搜索结果。

当搜索抵达大小减少的数据桶时，Splunk Web 上会显示一条消息，提醒用户此搜索的完成时间可能会有所延迟：“已完成对最新数据的搜索。对大小减少后数据桶的搜索速度可能会变慢。”

以下搜索命令不适用于大小减少后的数据桶：typeahead、tstats 和 walklex。当使用这些命令的搜索触及大小减少的数据桶时，search.log 中会新增一个警告：“完整大小的数据桶会返回搜索结果，但大小减少后的数据桶不会返回任何结果。”另外，对于 tstats 命令，Splunk Web 上会显示以下消息：“在 index={index} 中发现大小减少后的数据桶。Tstats 搜索不适用于大小减少后的数据桶。因此，搜索结果将是错误的。”

tsidx 减少不会触及加速数据模型的 tsidx 文件，这些文件保存在各自的目录中，且与索引数据桶分开。因此，仅作用于加速数据模型的 tstats 命令不会受此功能的影响，并将继续正常工作。

配置 tsidx 保存策略

默认情况下，索引器会保留数据桶生命周期内的所有 tsidx 文件。要更改这一策略，则必须启用 tsidx 减少。

Tsidx 的默认保存期为七天，您也可以修改这一默认值。只有当数据桶中的所有事件均超过保存期时，才会减少该数据桶的大小。

通过 Splunk Web 配置

要启用某索引的 tsidx 减少，编辑该索引：

1. 导航到**设置 > 索引**。
2. 单击要编辑的索引名称。
3. 转到“编辑”屏幕上的“存储优化”区域。
4. 在 “Tsidx 保存策略”字段，单击**启用减少**。
5. 要修改默认的保存期，编辑“减少超过以下时间的 tsidx 文件”字段。
6. 单击**保存**。

配置 indexes.conf

您可以直接编辑 indexes.conf 来启用 tsidx 减少。可以单独为一个或多个索引启用减少，也可以为所有索引进行全局启用。

要为单个索引启用 tsidx 减少，在 indexes.conf 中的该索引段落下放入相关属性。例如，要为 "newone" 索引启用减少并将保存期设为十天，则：

```
[newone]
enableTsidxReduction = true
timePeriodInSecBeforeTsidxReduction = 864000
```

要为所有索引启用 tsidx 减少，在 [default] 段落下放入相关设置。

必须重新启动索引器以使设置生效。

通过 CLI 配置

要为名为 "newone" 的索引启用 tsidx 减少并将保存期设为十天，则：

```
splunk edit index newone -enableTsidxReduction true -timePeriodInSecBeforeTsidxReduction 864000
```

运行此命令无需重新启动索引器。

首次启用 tsidx 减少时对性能的影响

在启用 tsidx 减少后，索引器即开始查找可以减少大小的数据桶。所有超过指定保存期的数据桶都会进行减少。索引器一次只会针对一个数据桶进行大小减少，因此对性能的影响应该很小。

确定数据桶的大小是否已减少

运行 dbinspect 搜索命令：

```
| dbinspect index=_internal
```

结果中的 tsidxState 字段说明了每个数据桶的大小是 "full" 还是 "mini"。

Tsidx 减少和索引器群集

索引器群集可以作为一个独立的索引器，以同样的方式且基于同样的规则和设置来运行 tsidx 减少。但是，由于只有可搜索的数据桶副本有 tsidx 文件可以用来减少，所以减少只针对可搜索的副本。在 tsidx 减少启用后，可搜索的数据桶副本可以保留最大大小的或迷你版的 tsidx 文件，具体取决于数据桶的时间。

必须采用**配置软件包**的方法将更改强行应用于 tsidx 减少设置。这可以确保所有的对等节点均采用同样的设置。随后，大小有待进行减少的数据桶的所有可搜索副本几乎同时进行 tsidx 减少，而不管他们处于哪个对等节点。

在数据桶的大小成功减少后，如果群集必须将此减少的数据桶的不可搜索副本转换为可搜索副本以满足搜索因子，则可采用以下两种转换方式：

- 如果群集中包含此数据桶的另一个可搜索副本，则此群集会将副本的迷你版 tsidx 文件流化为不可搜索的副本。在流化完成后，此副本则认为是可搜索的。
- 如果群集中不包含此数据桶的其他可搜索副本，则此群集没有可流化为不可搜索副本的迷你版 tsidx 文件。此时，群集必须先从不可搜索副本的原始数据文件中建立最大大小的 tsidx 文件，然后再减少完整文件。没有方式可以直接从原始数据文件创建迷你版 tsidx 文件。

有关索引器群集如何使数据桶的不可搜索副本变为可搜索副本的详细信息，请参阅“数据桶修复方案”。

将大小减少后的数据桶恢复到原始状态

您无法仅仅通过增加为 tsidx 减少设置的时间来将减少后的数据桶恢复到原始状态。该设置不会影响已减少的数据桶。

相反，要将数据桶的迷你版 tsidx 文件还原为最大大小的 tsidx 文件：

1. 停止索引器。

2. 在 indexes.conf 中，要么禁用 tsidx 减少，要么增加 tsidx 减少的时间设置，使该时间大于您要恢复的数据桶的时间。否则，数据桶会在您恢复文件后再次进行大小减少。

3. 在数据桶上运行 splunk rebuild 命令：

```
splunk rebuild <bucket directory><index name>
```

请参阅“重新构建单个数据桶”。

4. 重新启动索引器。

tsidx 减少的限制

SmartStore

您不能减少 **SmartStore** 索引中数据桶的 tsidx 文件。但是，在迁移到 SmartStore 之前，您仍然可以搜索任何已减少 tsidx 的搜索数据桶。请参阅“关于 SmartStore”。

索引实时搜索

Tsidx 减少与已建立索引的实时搜索不兼容。另外，对于依赖已建立索引的实时搜索的应用程序，不应在索引上启用 tsidx 减少功能；例如，ITSI 中的索引 itsit_tracked_alerts。

Tsidx 写入级别

Splunk 平台使用时间序列索引文件来维护关键字列表，以便在搜索数据时加快索引数据桶的选择。默认情况下，索引器只要保留索引数据，就会一直保留 tsidx 文件。tsidx 文件对于有效搜索大量数据非常重要，但同时它们也需占用大量存储空间。通过调整 tsidx 写入级别，您可以减少维护 tsidx 文件对总体存储空间的影响。

有多种优化措施可用于改善 tsidx 压缩。这些优化封装在各个级别中，更高版本的 Splunk Enterprise 中添加了新的级别。更改 indexes.conf 中的默认的 tsidxWritingLevel 设置会更改索引 tsidx 文件和数据模型加速所使用的优化。请参阅*《管理员手册》*中的 indexes.conf。

tsidxWritingLevel 的兼容性

更高版本的 Splunk Enterprise 中添加了新的级别，因此您可以使用进一步改进的优化功能。将 tsidxWritingLevel 设置为最高级别会限制搜索数据的向后兼容性。

Splunk Enterprise 版本	支持的 tsidxWritingLevel	可用于搜索的 tsidx 级别	默认值
7.2.x	1、2	1、2	1
7.3.x	1、2、3	1、2、3	1
8.0.x	1、2、3	1、2、3	1
8.1.x	1、2、3、4	1、2、3、4	1
8.2.x	1、2、3、4	1、2、3、4	2

示例：由配置为 tsidxWritingLevel=3 的 Splunk Enterprise 8.1.x 索引器写入的索引数据桶可以在 Splunk Enterprise 7.3.x 上搜索。由配置为 tsidxWritingLevel=4 的 Splunk Enterprise 8.2.x 索引器写入的索引数据桶无法在低于 8.1.x 版本的 Splunk Enterprise 中搜索。

如果您的基础架构规划要求向后兼容较低的主要 Splunk Enterprise 版本，请选择较低的 tsidxWritingLevel。

更改设置后的预期行为

在将索引器或群集对等节点升级到更高版本的 Splunk Enterprise 之后，建议查看 tsidxWritingLevel 并将其调整到更高的级别。

- 对 tsidxWritingLevel 的更改将应用于新的索引数据桶 tsidx 文件。现有的 tsidx 文件不会有任何更改。
- 对 tsidxWritingLevel 的更改将应用于新加速的数据模型，或者在重建现有数据模型启动后进行应用。所有现有的数据模型加速将不受影响。

若想在升级多站点索引器群集时不会出现搜索中断，请将更改 tsidxWritingLevel 的时间推迟到多站点索引器群集升级完成之后。配置所有对等节点，使所有节点同时切换到新的级别。

合并 tsidx 设置

tsidxWritingLevel 设置可以与 tsidx 保留策略结合使用，以显著降低值较高、可立即搜索的数据所使用的存储空间以及该数据的长期存储影响。

确定 indexes.conf 的哪种更改需要重新启动

一些对 indexes.conf 的更改需要重新启动索引器以使更改生效：

- 更改如下任意属性： rawChunkSizeBytes, minRawFileSyncSecs, syncMeta, maxConcurrentOptimizes, coldToFrozenDir, coldtoFrozenScript
- 为现有索引更改如下任意属性： repFactor, homePath, coldPath, thawedPath, bloomHomePath, summaryHomePath,

tstatsHomePath, remotePath, coldPath.maxDataSizeMB, datatype

- 添加或删除卷
- 启用或禁用包含数据的索引
- 删除一个索引

如果仅作如下更改，则不必重新启动索引器：

- 添加新的索引段落。
- 更改任何未列于“需重新启动”之列的属性
- 启用或禁用不含数据的索引

导致索引器群集中对等节点进行滚动重新启动的配置更改是本文所列更改的超集。请参阅“配置软件包更改后重新启动或重新加载？”

注意：关于除 indexes.conf（需要重新启动）以外的其他配置更改的信息，请参阅*《管理员手册》*中的“配置文件更改后何时重新启动 Splunk Enterprise”。

使用监视控制台查看索引和卷的状态

您可以使用监视控制台监视部署的大多数方面。本主题介绍可用来深入了解索引性能的控制台仪表板。

监视控制台的主要文档位于*《监视 Splunk Enterprise 手册》*内。

有若干个监视索引和卷的状态的仪表板。仪表板的使用范围或者是单个实例，或者是整个部署。它们位于**索引**菜单下方：

- 索引和卷：实例
- 索引和卷：部署
- 索引详细信息：实例
- 索引详细信息：部署
- 卷详细信息：实例
- 卷详细信息：部署

这些仪表板提供了关于索引和卷的丰富信息，例如：

- 按索引的磁盘使用情况
- 卷使用情况
- 索引和卷大小随时间的变化情况
- 数据时间
- 数据桶类型的统计信息
- 数据桶设置

有关更多信息，请查看仪表板本身。此外，您还可以参阅*《监视 Splunk Enterprise》*中的“建立索引：索引和卷”。

备份和归档您的索引

备份索引数据

注意：本主题大部分内容都与 **SmartStore** 索引无关。请参阅“关于 SmartStore”。

要确定如何备份已索引数据，最好先了解索引器如何存储数据以及在创建数据索引后数据是如何老化的。然后再决定备份策略。

阅读本主题之前，应先阅读“索引器如何存储索引”，以便对索引的结构以及配置索引的方法有个大致的了解。不过，如果您想要立即开始阅读本主题，下一部分有尝试汇总“索引器如何存储索引”主题中的若干要点。

数据如何老化

已索引的数据会驻留在包含称为**数据桶**的子目录的数据库目录中。每个索引都有其自己的一组数据库。

数据老化时，会经历多种类型的数据桶。您可以通过配置 indexes.conf 中的属性来决定数据如何老化。有关控制数据如何老化的 indexes.conf 中的设置介绍，请参阅“配置索引存储”。

下面是关于数据如何在索引中老化的简单概括：

1. 当索引器首次为数据创建索引时，数据进入“热”数据桶。一次可有多个热数据桶处于打开状态，具体取决于您的配置。由于索引器主动向热数据桶中写入内容，因此无法备份热数据桶，但您可对其拍摄快照。

2. 数据将停留在热数据桶中，直到满足可将数据重新分类为“温”数据的策略条件。这项操作称为将数据“滚动”到温数据桶中。当热数据桶达到指定大小或时间或者重新启动 splunkd 时，将会发生这种情况。滚动热数据桶时，将对其目录重命名，然后成为温数据桶。（您也可以按照下文中的介绍，手动将热数据桶滚动到温数据桶。）对温数据桶进行备份是安全的。

3. 当索引达到其中某一项可能的可配置限制时（通常为指定的温数据桶数量），时间最长的数据桶将变为“冷”数据桶。索引器将该数据桶移动到 colddb 目录。温数据桶的默认数量为 300。

4. 最后，根据您定义的策略要求中所规定的时间，数据桶将从冷数据桶滚动到“冻结”数据桶。索引器会删除冻结数据桶。但是，如果您需要保留此数据，可以让索引器在删除数据桶之前对数据进行归档。有关更多信息，请参阅“归档索引的数据”。

您可以通过控制多个不同的参数（如索引或数据桶的大小或数据的时间）来设置退休和归档策略。

总结：

- **热数据桶** - 当前正向其写入信息的数据桶；不对其进行备份。
- **温数据桶** - 从热数据桶滚动而来；可安全备份。
- **冷数据桶** - 从温数据桶滚动而来；这些数据桶会被移动到另一个位置。
- **冻结数据桶** - 索引器会删除这些数据桶，但您可以在此之前先对其内容进行归档。

您可以在 indexes.conf 中设置索引数据库的位置。（请参阅下文了解有关默认索引的数据库位置的详细信息。）您还可以在此文件中指定许多其他属性，例如，热数据桶的最大大小以及时间。

索引数据库目录的位置

下面是默认索引（defaultdb）的目录结构：

数据桶类型	默认位置	注释
热	$SPLUNK_HOME/var/lib/splunk/defaultdb/db/*	可存在多个热子目录。每个热数据桶都会占用自己的子目录，它使用以下命名约定： hot_v1_<ID>
温	$SPLUNK_HOME/var/lib/splunk/defaultdb/db/*	每个温数据桶都有单独的子目录。这些子目录将按下文“温/冷数据桶命名约定”中的介绍命名。
冷	$SPLUNK_HOME/var/lib/splunk/defaultdb/colddb/*	有多个冷子目录。当温数据桶滚动到冷数据桶时，它们会移动到此目录中，但不会重新命名。
冻结	N/A：冻结数据将被删除或者归档到您指定的目录位置中。	默认设置为删除冻结数据；有关如何改为归档冻结数据的信息，请参阅“归档索引的数据”。
解冻	$SPLUNK_HOME/var/lib/splunk/defaultdb/thaweddb/*	曾经归档后来又被解冻的数据的位置。有关将归档数据恢复到解冻状态的信

解冻	$SPLUNK_HOME/var/lib/splunk/defaultdb/thaweddb/*	息，请参阅“恢复归档的数据”。

热/温和冷目录的路径是可配置的，因此可将冷数据桶存储在不同于热/温数据桶的单独位置中。请参阅“配置索引存储”和“对索引数据使用多个分区”。

重要提示：所有索引位置必须是可写位置。

选择备份策略

需要考虑两种基本备份方案：

- 持续、增量的温数据备份
- 备份所有数据–例如，在升级索引器之前

当然，实际执行备份的方式将完全取决于您组织中准备就绪的工具和过程，但本部分将向您提供继续操作所需的指导原则。

增量备份

常规建议是使用您选择的增量备份实用工具定期计划所有新的温数据桶的备份。如果需要频繁滚动数据桶，还应该将冷数据库目录包括在您的备份中，确保不会错过已滚动到冷状态的所有尚未备份的数据桶。由于数据桶从温滚动到冷时，数据桶目录名称不会变化，因此，您可以仅按名称过滤。

如果还要备份热数据桶，您需要使用类似 VSS（在 Windows/NTFS 上）、ZFS 快照（在 ZFS 上）等工具，或使用存储子系统提供的快照实用工具拍摄文件的快照。如果您没有可用的快照工具，可以按照下文所述，手动将热数据桶滚动到温数据桶，然后对其进行备份。但是，这并不是一般建议，具体原因下文中也有说明。

备份所有数据

在升级索引器之前，建议您备份所有数据。也就是说，备份热数据桶、温数据桶和冷数据桶。

很显然，可以通过很多方式来执行此操作，具体操作方式将取决于您数据的大小以及您能够承担多少停机时间。以下是一些基本指导原则：

- 如果数据量较小，可关闭索引器并只对数据库目录进行备份，然后执行升级。
- 如果数据量较大，可能需要创建热数据桶的快照，然后再执行升级。

在任何情况下，如果您一直在对从热数据桶滚动而来的温数据桶执行增量备份，那么此时您确实只需备份您的热数据桶。

手动将热数据桶滚动到温数据桶

要手动将索引的热数据桶滚动到温数据桶，可使用以下 CLI 命令，并将 `<index_name>` 替换为您要滚动的索引的名称：

```
splunk _internal call /data/indexes/<index_name>/roll-hot-buckets -auth <admin_username>:<admin_password>
```

重要提示：通常不建议手动滚动热数据桶，因为每次强制滚动都会永久降低对数据搜索的性能。正常情况下，较大的数据桶的搜索效率会更高。过早地对数据桶执行滚动操作，将会生成较小且效率较低的数据桶。如果需要对热数据进行备份，快照备份是首选的方法。

注意：在利用加速数据模型摘要的环境中，不建议手动滚动热数据桶。如果在热数据桶滚动的同时有其他索引管理任务正在进行，则可能会危及数据的完整性。

关于恢复的建议

如果您遇到非灾难性的磁盘故障（例如，您的某些数据还存在，但索引器无法运行），Splunk 建议您将索引目录移动到一边，然后从备份中进行恢复，而不是在已经部分损坏的数据存储上进行恢复。索引器在启动时会自动创建热目录并继续创建索引。监视的文件和目录将从备份时的位置开始。

群集数据备份

即使**索引器群集**已经包含冗余数据副本，您可能还希望将群集数据备份到另一个位置；例如，备用保存一份数据副本作为整体灾难恢复计划的一部分。

实现此目的最简单的方法是，按照本主题上文中所述，使用在个别非群集索引器上备份数据的相同方法来备份在群集的各个对等节点上的数据。但是，这种方法会对复制的数据进行备份。例如，如果某群集的**复制因子**为 3，该群集将在其对等节点集上存储所有数据的三个副本。如果您对驻留每个个别节点上的数据进行备份，那么您最终获得的备份将总共包含数据的三个副本。如果只备份单个节点上的数据，则无法解决这个问题，因为无法确定单个节点是否包含群集中的所有数据。

该问题的解决方案是，先找出群集上每个数据桶的一个副本，然后只对这些副本进行备份。但是，该解决方案在实际操作过程中是非常复杂的。一个办法是，创建一个脚本，让该脚本遍历每个对等节点的索引存储，然后使用数据桶名称中包含的数据桶 ID 值来找出每个数据桶的一个副本。数据桶的所有副本与数据桶 ID 都是相同的。有关数据桶 ID 的信息，请参阅“温/冷数据桶命名约定”。设计群集备份脚本时还需要考虑的是，您是否只想备份数据桶的原始数据，还是要同时备份其原始数据和索引文件。对于后面这种情况，脚本还需要找出每个数据桶的可搜索副本。

由于群集备份非常复杂，因此建议您与 Splunk 专业服务联系，以便在对群集数据的单个副本进行备份时获得指导。他们还可以根据您的环境需要，为您定制一个解决方案。

设置退休和归档策略

注意：本主题大部分内容都与 **SmartStore** 索引无关。请参阅“为 SmartStore 索引配置数据保留”。

您可以通过控制索引的大小以及索引中数据的时间来配置数据退休和**归档**策略。

索引器将索引数据存储在名为**数据桶**的目录中。数据桶会经历四个退休阶段。索引数据到达最终的冻结状态时，索引器会从索引中将其删除。您可以将索引器配置为在数据冻结时对数据归档，而不是将其完全删除。有关详细信息，请参阅“归档索引的数据”。

数据桶阶段	描述	是否可搜索？
热	包含新建索引的数据。允许写入。每个索引存在一个或多个热数据桶。	是
温	从热数据桶滚动来的数据。温数据桶会有许多个。	是
冷	从温数据桶滚动来的数据。有许多个冷数据桶。	是
冻结	从冷数据桶滚动来的数据。默认情况下，索引器将删除冻结的数据，但您也可将其归档。可在以后将已归档数据解冻。	否

您将通过编辑 indexes.conf 来配置索引及其数据桶的大小、位置和时间，如“配置索引存储”中所述。

警告：您更改数据退休和归档策略设置时，索引器可能会删除旧数据，而不会发出任何提示。

为冷到冻结的滚动行为设置属性

maxTotalDataSizeMB 和 frozenTimePeriodInSecs 属性（位于 indexes.conf 中）可帮助确定何时将冷数据桶滚动到冻结数据桶。下文对这些属性进行了详细介绍。

当索引变得太大时冻结数据

可以使用索引的大小来确定何时将数据冻结并从索引中删除。如果索引大小超过指定的最大大小，时间最长的数据将滚动到冻结状态。

索引的默认最大大小为 500,000MB。要更改最大大小，编辑 maxTotalDataSizeMB 属性（位于 indexes.conf 中）。例如，可以将最大大小指定为 250,000MB：

```
[main]
maxTotalDataSizeMB = 250000
```

请以兆字节为单位指定此大小。

重新启动索引器后，新设置才会生效。根据要处理的数据量而定，索引器将数据桶移出索引以便符合新策略需要花费一定的时间。在此过程中，您可能发现 CPU 使用率非常高。

此设置与 frozenTimePeriodInSecs 一起用于确定何时冻结数据。达到任一设置时，数据都会冻结。

如果在 frozenTimePeriodInSecs 之前达到了 maxTotalDataSizeMB，则数据将在配置的时间周期过去之前滚动到冻结状态。如果尚未正确配置归档策略，数据可能会意外丢失。

当数据时间太久时将其冻结

可以使用数据的时间来确定数据桶何时滚动到冻结数据桶。当特定数据桶中最近的数据到达配置的时间时会滚动整个数据桶。

如需指定数据冻结的时间，请编辑 frozenTimePeriodInSecs 属性（位于 indexes.conf 中）。该属性指定冻结数据之前所经历的秒数。默认值为 188697600 秒，或者约为 6 年。以下示例将对索引器进行配置，以便在旧事件超过 180 天（15552000 秒）时

从索引中将其挑选出来：

```
[main]
frozenTimePeriodInSecs = 15552000
```

请以秒为单位指定时间。

重新启动索引器后，新设置才会生效。根据要处理的数据量而定，索引器将数据桶移出索引以便符合新策略需要花费一定的时间。在此过程中，您可能发现 CPU 使用率非常高。

归档数据

如果希望将冻结数据归档，而不是将其整个删除，您必须按“归档索引的数据”中的介绍告知索引器执行此操作。您可以创建自己的归档脚本，也可以就让索引器为您完成归档操作。之后，您可以按“恢复归档的数据”中所述恢复（“解冻”）归档的数据。

数据桶老化的其他方式

还有许多其他条件会导致数据桶从一个阶段滚动到另一个阶段，其中的某些条件也会触发删除或归档行为。这些条件都是可配置的，如“配置索引存储”所述。要完整了解控制退休策略的所有选项，请参阅该主题并查看 indexes.conf 规范文件。

例如，索引器会在数据桶达到其最大大小时滚动数据桶。您可以设置较小的 maxDataSize（位于 indexes.conf 中）来减小数据桶大小，以便加快数据桶的滚动速度。但要注意的是，搜索较小的数据桶所花费的时间比搜索少量较大的数据桶所花费的时间要长。要获得所需的结果，您必须试验若干次，以便确定数据桶的最佳大小。

故障排除归档策略问题

我的磁盘空间不足了，因此我更改了归档策略，但仍然不能正常运行。

如果在磁盘空间不足的情况下，您将归档策略更改得更加严格，您可能发现并未开始按照您的新策略对事件进行归档。最可能的原因是，您必须先释放一些空间来为进程提供运行空间。停止索引器，清理出大约 5GB 的磁盘空间，然后再次启动索引器。一段时间后（具体时间将取决于要处理的数据量），您应看到与 BucketMover（位于 splunkd.log 中）相关的 INFO 条目，显示正在对数据桶进行归档。

归档索引的数据

注意：尽管 **SmartStore** 索引通常不包含冷数据桶，当数据桶直接从温状态滚动到冻结状态时，您仍可使用本文介绍的属性（coldToFrozenDir 和 coldToFrozenScript）归档 SmartStore 数据桶。请参阅“为 SmartStore 索引配置数据保留”。

您可以将索引器配置为在数据老化时自动对其进行归档，具体来说，就是在它滚动到“冻结”状态时。为此，您需要配置 indexes.conf。

警告：默认情况下，索引器将删除所有冻结的数据。它会在数据变为冻结状态时立即从索引中将其删除。如果您需要保留数据，则**必须**将索引器配置为在删除数据之前对其进行归档。为此，您可以设置 coldToFrozenDir 属性或指定一个有效的 coldToFrozenScript（位于 indexes.conf 中）。

有关数据存储的详细信息，请参阅“索引器如何存储索引”。有关编辑 indexes.conf 的信息，请参阅配置索引存储。

索引器如何归档数据

索引器根据数据退休策略，把旧数据从索引中旋转出来，如“设置退休和归档策略”中所述。数据会经历若干个阶段，这些阶段与文件目录位置对应。数据从**热**数据库开始，该数据库位于 $SPLUNK_HOME/var/lib/splunk/defaultdb/db/ 下的子目录（“**数据桶**”）中。然后，它移动到**温**数据库，同样位于 $SPLUNK_HOME/var/lib/splunk/defaultdb/db 下的子目录中。最终，数据老化，进入**冷**数据库 $SPLUNK_HOME/var/lib/splunk/defaultdb/colddb。

最后，数据到达**冻结**状态。出现这种情况的原因很多，详参“设置退休和归档策略”。此时，索引器将清除索引中的数据。如果希望索引器在清除索引中的冻结数据之前将其存档，则必须指定该行为。您可使用以下两种方式处理归档：

- 让索引器自动执行归档。
- 让索引器运行指定的归档脚本。

归档行为取决于您设置了这些 indexes.conf 中的哪个属性：

- coldToFrozenDir. 此属性指定索引器用于自动归档冻结数据的位置。
- coldToFrozenScript. 此属性指定冻结数据时，索引器将运行的脚本。通常，该脚本将是用于归档冻结数据的脚本。该脚本也可用作其他用途。索引器随附了一个可编辑和使用的示例归档脚本（$SPLUNK_HOME/bin/coldToFrozenExample.py），但您可以实际指定希望索引器运行的任何脚本。

注意：您只能设置这两个属性之一。如果同时设置这两个属性，coldToFrozenDir 属性将优先于 coldToFrozenScript。

如果您没有指定任何属性，索引器将运行默认脚本，只是将要清除数据桶的名称写入日志文件 $SPLUNK_HOME/var/log/splunk/splunkd_stdout.log。然后清除该数据桶。

让索引器为您归档数据

如果设置了coldToFrozenDir 属性（位于 indexes.conf 中），索引器会在清除索引中的数据之前自动将冻结数据桶复制到指定位置。

将此段落添加到 $SPLUNK_HOME/etc/system/local/indexes.conf：

```
[<index>]
coldToFrozenDir = <path to frozen archive>
```

请注意以下事项：

- <index> 指定哪个索引包含要归档的数据。
- <path to frozen archive> 指定索引器用于放置已归档数据桶的目录。

注意：使用 Splunk Web 创建新索引时，也可以为该索引指定一个冻结归档路径。详细信息请参阅“创建自定义索引”。

索引器如何归档冻结数据取决于最初为数据创建索引时是否使用的是 4.2 之前的版本：

- 对于使用 4.2 版及更高版本所创建的数据桶，索引器将删除**原始数据文件**以外的所有文件。
- 对于使用 4.2 之前版本所创建的数据桶，脚本将只使用 gzip 压缩数据桶中的所有 .tsidx 和 .data 文件。

存在此差异的原因是原始数据的格式发生了变化。从 4.2 版本开始，原始数据文件包含了索引器重组索引数据桶所需的所有信息。

有关解冻这些数据桶的信息，请参阅“恢复归档的索引数据”。

指定归档脚本

如果您设置了coldToFrozenScript 属性（位于 indexes.conf 中），您指定的脚本将在索引器从索引中清除冻结数据之前运行。

您需要提供实际脚本。通常，脚本将归档数据，但您可以提供一个执行任何所需操作的脚本。

将此段落添加到 $SPLUNK_HOME/etc/system/local/indexes.conf：

```
[<index>]
coldToFrozenScript = ["<path to program that runs script>"] "<path to script>"
```

请注意以下事项：

- <index> 指定哪个索引包含要归档的数据。
- <path to script> 指定归档脚本的路径。该脚本必须位于 $SPLUNK_HOME/bin 或它的某个子目录中。
- <path to program that runs script> 为可选。如果脚本需要通过某个程序（如 python）来运行，则必须指定此选项。
- 如果脚本位于 $SPLUNK_HOME/bin 且名称为 myColdToFrozen.py，可按如下方式设置该属性：

```
coldToFrozenScript = "$SPLUNK_HOME/bin/python" "$SPLUNK_HOME/bin/myColdToFrozen.py"
```

- 有关归档脚本的详细信息，请参阅 indexes.conf 规范文件。

索引器随附一个可供您编辑的归档脚本示例，即 $SPLUNK_HOME/bin/coldToFrozenExample.py。

注意：如果使用此脚本示例，可以对其进行编辑，以指定安装的归档位置。此外，还要重命名该脚本或将其移动到另一个位置，以避免升级索引器时更改内容被覆盖。**这是一个脚本示例**，因此在尚未根据您的环境进行相应编辑并广泛测试的情况下，不能将其套用于生产实例。

脚本示例对冻结数据的归档方式有所不同，这取决于最初为数据创建索引时是否使用的是 4.2 之前的版本：

- 对于使用 4.2 版及更高版本所创建的数据桶，它将删除原始数据文件以外的所有文件。
- 对于使用 4.2 之前版本的数据桶，脚本将只使用 gzip 压缩所有 .tsidx 和 .data 文件。

存在此差异的原因是原始数据的格式发生了变化。从 4.2 版本开始，原始数据文件包含了索引器重组索引数据桶所需的所有信息。

有关解冻这些数据桶的信息，请参阅“恢复归档的索引数据”。

作为最佳做法，请确保您创建的脚本可尽快地完成，以便索引器不必等待传回指示器。例如，如果想要归档到一个较慢的卷上，则将脚本设置为将数据桶复制到与索引相同的（快）卷上的某个临时位置。然后，在索引器之外，使用单独脚本将该临时位置的数据桶移动到较慢卷上的目标位置。

数据归档和索引器群集

在**索引器群集**中，每个单独对等节点将它的数据桶滚动到冻结状态，按照与非群集索引器相同的方式进行；即，基于它自己的配置组进行。因为群集中所有对等节点的配置都应该相同，所以数据桶的所有副本都应该几乎在同一时间滚动到冻结状态。

然而，在时间上可能会有一些差异，因为相同的索引在不同的对等节点上可能以不同的速率增长。群集会执行处理，以确保数据桶在群集中的所有对等节点之间顺利地冻结。具体地说，它会执行处理，这样如果一个数据桶在一个对等节点上冻结了而未在另一个对等节点上冻结，群集不会启动该数据桶的修复活动。请参阅“群集如何处理冻结数据桶”。

归档多个副本的问题

因为索引器群集包含每个数据桶的多个副本，如果您使用本主题前面介绍的方法归档数据，您将归档数据的多个副本。

例如，如果某群集的**复制因子**为 3，该群集将在其对等节点集上存储所有数据的三个副本。如果您将每个对等节点设置为在自己的数据滚动到冻结状态时对其进行归档，您最终将获得数据的三个归档副本。如果只归档单个节点上的数据，则无法解决这个问题，因为无法确定单个节点是否包含群集的所有数据。

该问题的解决方案是，归档群集上每个数据桶的一个副本，并放弃其他副本。但是，该解决方案在实际操作过程中是非常复杂的。如果在归档群集数据的单个副本时需要相关指导，请与 Splunk 专业服务联系。他们还可以根据您的环境需要，为您定制一个解决方案。

指定归档目标

如果您选择采用简单的方式归档群集数据的多个副本，则必须防止出现命名冲突。您不能将所有对等节点上的数据全都发送到单一归档目录中，因为，群集内将存在数据桶的多个采用相同方式命名的副本（对于复制因子 > 2 的部署），而目录中所含内容的名称必须唯一。您需要确保将每个对等节点的数据桶传输到一个单独的归档目录。当然，如果您通过 `coldToFrozenDir` 属性（位于 `indexes.conf` 中）指定了共享存储中的某个目标目录，这在管理上将存在一点困难，因为如“在索引器群集中配置对等节点索引”中所述，`indexes.conf` 文件在所有对等节点上必须是相同的。另一种方法是，创建一个脚本，让该脚本将每个对等节点的归档数据桶传输到共享存储上的一个单独位置，然后使用 `coldToFrozenScript` 属性指定该脚本。

恢复归档的索引数据

您可以通过将归档的**数据桶**移动到解冻目录（如 `$SPLUNK_HOME/var/lib/splunk/defaultdb/thaweddb`）并在之后对其进行处理（将在本主题的后文中介绍）来恢复归档的数据。`thaweddb` 中的数据不易受服务器的索引老化方案（热 > 温 > 冷 > 冻结）影响。您可以根据需要在解冻目录中将归档数据保留任意长的时间。当您不再需要此数据时，只需将其删除或从解冻目录中移出即可。

重要提示：您恢复归档数据的方式会有所不同，这取决于最初为数据创建索引时是否使用的是 4.2 或更高版本的 Splunk Enterprise。这是因为 Splunk Enterprise 4.2 版更改了原始数据的格式。

有关如何首先归档数据的信息，请参阅“归档索引的数据”。如果在解冻数据后想要重新对其进行归档，也可以使用该页面作为指导。

恢复的数据不计入许可证。

将归档恢复到另一个索引器实例的限制

在大多数情况下，可以将归档恢复到任何 Splunk 实例，而不仅仅是最初为其建立索引的那个索引器实例。但是，这取决于多种因素：

- **Splunk Enterprise 版本。**您不能将 Splunk Enterprise 4.2 或更高版本所创建的数据桶恢复到 4.2 版之前的索引器。4.1 与 4.2 的数据桶数据格式有所更改，而且 4.2 版之前的索引器不了解新格式。也就是说：
  - **4.2 版本以上的数据桶：**您可以将 4.2 版本以上的数据桶恢复到任何 4.2 版本以上的实例。
  - **4.2 版之前的数据桶：**除了下一项目符号中介绍的一些与操作系统相关的问题以外，您可以将 4.2 版之前的数据桶恢复到 4.2 版之前或之后的任何索引器。
- **操作系统版本。**您可以将数据桶恢复到另一操作系统上运行的索引器。具体来说：
  - **4.2 版本以上的数据桶：**您可以将 4.2 版本以上的数据桶恢复到运行在任何操作系统上的索引器。
  - **4.2 版之前的数据桶：**您可以将 4.2 版之前的数据桶恢复到运行在任何操作系统上的索引器，唯一的限制是您不能将 4.2 版之前的数据恢复到具有不同端序的系统。例如，在 64 位元系统上生成的数据将无法在 32 位元系统上

正常运行，也不能将数据从 PowerPC 或 Sparc 系统移动到 x86 或 x86-64 系统，反之亦然。

此外，请确保在恢复归档的数据桶时未将数据桶 ID 冲突引入索引中。此问题将在下文中讨论。

如何了解归档数据桶是否包含 4.2+ 数据

解冻归档数据桶之前，您需要确定归档数据桶是 4.2 之前还是 4.2 之后的版本。下面介绍两者的不同之处，假定您使用 coldToFrozenDir 或提供的脚本示例归档了数据桶：

- **4.2 版本以上数据桶：**数据桶目录只包含原始数据目录，该目录包含 journal.gz。
- **4.2 版之前的数据桶：**数据桶目录包含 .tsidx 和 .data 文件的 gzip 压缩版本以及含有名为 <int>.gz 文件的原始数据目录。

重要提示：如果您是通过自己的脚本归档数据，那么生成的数据桶可能包含任何内容。

如果您使用 coldToFrozenDir 或提供的脚本示例归档了数据桶，则可以按照以下步骤将其解冻。

解冻 4.2 版本以上归档

**nix 用户*

以下是将 4.2 版本以上归档数据桶安全恢复到解冻状态的一个示例：

1. 将您的归档数据桶复制进解冻目录：

```
cp -r db_1181756465_1162600547_1001 $SPLUNK_HOME/var/lib/splunk/defaultdb/thaweddb
```

注意：数据桶 id 不能和索引中的任何其他数据桶发生冲突。本示例假设数据桶 id '1001' 在索引中是唯一的。如果不是，请选择其他不冲突的数据桶 ID。

2. 对归档数据桶执行 splunk rebuild 命令，重新构建索引以及关联的文件：

```
splunk rebuild $SPLUNK_HOME/var/lib/splunk/defaultdb/thaweddb/db_1181756465_1162600547_1001
```

3. 重新启动索引器：

```
splunk restart
```

Windows 用户

以下是将 4.2 版本以上归档数据桶安全恢复到解冻状态的一个示例：

1. 将您的归档数据桶复制进解冻目录：

```
xcopy D:\MyArchive\db_1181756465_1162600547_1001 %SPLUNK_HOME%\var\lib\splunk\defaultdb\thaweddb\db_1181756465 _1162600547_1001
/s /e /v
```

注意：数据桶 id 不能和索引中的任何其他数据桶发生冲突。本示例假设数据桶 id '1001' 在索引中是唯一的。如果不是，请选择其他不冲突的数据桶 ID。

2. 对归档数据桶执行 splunk rebuild 命令，重新构建索引以及关联的文件：

```
splunk rebuild %SPLUNK_HOME%\var\lib\splunk\defaultdb\thaweddb\db_1181756465_1162600547_1001
```

3. 重新启动索引器：

```
splunk restart
```

解冻 4.2 版之前的归档

**nix 用户*

以下是将 4.2 版之前的归档数据桶安全恢复到解冻状态的一个示例：

1. 将您的归档数据桶复制到解冻目录中的某个临时位置：

```
# cp -r db_1181756465_1162600547_0  $SPLUNK_HOME/var/lib/splunk/defaultdb/thaweddb/temp_db_1181756465_1162600547_0
```

2. 如果最初归档时对数据桶进行了压缩，则在解冻目录中将内容解压缩。

3. 将临时数据桶重命名为索引器可以识别的名称：

```
# cd $SPLUNK_HOME/var/lib/splunk/defaultdb/thaweddb/
# mv temp_db_1181756465_1162600547_0 db_1181756465_1162600547_1001
```

注意：您必须选择不会与索引中的任何其他数据桶发生冲突的数据桶 id。本示例假设数据桶 id '1001' 在索引中是唯一的。如果不是，请选择其他不冲突的数据桶 ID。

4. 刷新清单：

```
# cd $SPLUNK_HOME/bin
# ./splunk login
# ./splunk _internal call /data/indexes/main/rebuild-metadata-and-manifests
```

片刻之后，新解冻的数据桶中的内容便恢复可搜索状态。

Windows 用户

以下是将 4.2 版之前的归档数据桶安全恢复到解冻状态的一个示例：

1. 将您的归档数据桶复制到解冻目录：

```
> xcopy D:\MyArchive\db_1181756465_1162600547_0 %SPLUNK_HOME%\var\lib\splunk\defaultdb\thaweddb\temp_db_1181756465
_1162600547_0 /s /e /v
```

2. 如果最初归档时对数据桶进行了压缩，则在解冻目录中将内容解压缩。

3. 将临时数据桶重命名为索引器可以识别的名称：

```
> cd %SPLUNK_HOME%\var\lib\splunk\defaultdb\thaweddb
> move temp_db_1181756465_1162600547_0 db_1181756465_1162600547_1001
```

注意：您必须选择不会与索引中的任何其他数据桶发生冲突的数据桶 id。本示例假设数据桶 id '1001' 在索引中是唯一的。如果不是，请选择其他不冲突的数据桶 ID。

4. 刷新清单：

```
> cd %SPLUNK_HOME%\bin
> splunk login
> splunk _internal call /data/indexes/main/rebuild-metadata-and-manifests
```

片刻之后，新解冻的数据桶中的内容便恢复可搜索状态。

群集数据解冻

您可以按照将数据解冻到任何个别索引器的相同方法将归档的群集数据解冻到个别对等节点。但是，如“归档索引的数据”中所述，只首先归档群集数据的单个副本是很困难的。如果换一种方式，您先归档群集中所有对等节点的数据，可以之后再解冻数据，将数据放入最初归档该数据的对等节点的解冻目录中。最终您将获得群集上解冻数据的复制因子的副本，因为您解冻了所有原始数据，包括副本。

注意：解冻目录中的数据不会进行复制。因此，如果您只解冻了一些数据桶的一个副本，而不是所有副本，则只有一个副本会驻留在群集中，即您将数据放置到的对等节点的解冻目录中。

SmartStore 索引和数据桶解冻

有关解冻 SmartStore 索引中的归档数据的信息，请参阅“解冻数据和 SmartStore”。

索引器群集和索引复制概述

关于索引器群集和索引复制

索引器群集是配置为复制彼此数据的一组 Splunk Enterprise 索引器，这样系统便会保留所有数据的多个副本。此过程称为**索引复制**。通过保留 Splunk Enterprise 数据的多个相同副本，群集能够阻止数据丢失，同时还便于数据搜索。

索引器群集功能会自动从一个索引器故障转移到下一个索引器。这意味着，如果一个或多个索引器出现故障，可继续为传入数据新建索引，且索引数据继续保持可搜索状态。

索引复制的重要益处包括：

- **数据可用性**。始终有一个索引器可用于处理传入的数据，可以对索引数据进行搜索。
- **数据保真度**。您永远不会丢失任何数据。您可以确保发送到群集的数据与群集中存储的数据完全相同，并且之后可以对此数据进行搜索。
- **数据恢复**。您的系统可以容许发生故障索引器，而不会出现丢失数据或无法访问数据的情况。
- **灾难恢复**。有了多站点群集化，您的系统可容许整个数据中心的故障。
- **搜索相关性**。有了多站点群集化，搜索头能通过本地站点访问整个数据组，大大降低了长距离网络流量。

索引复制的关键问题是需要在数据可用性/恢复所带来的益处与存储成本（以及从较小程度而言增加的处理负载）之间找到平衡点。群集所处理的数据恢复的程度与它所保留的数据的副本数成正比。但是，保留更多的副本数意味着更高的存储要求。要做好权衡以满足您企业的需求，您可以对群集保留的副本数进行配置。这就是所谓的**复制因子**。

注意：具有 **SmartStore** 索引的群集依赖于远程对象存储的继承功能确保高可用性、数据保真度、数据恢复和大部分索引数据的灾难恢复。请参阅“关于 SmartStore”。

您还可以使用群集来调整索引容量，即使在没有对索引复制做出要求的情况下也是如此。请参阅“使用索引器群集调整索引”。

注意：搜索头群集为搜索头组提供高可用性和可扩展性。这些是一个独立于索引器群集的单独功能，但可以将它们与索引器群集结合使用，以便在整个 Splunk Enterprise 部署中构建一个高可用性、可扩展的解决方案。请参阅*《分布式搜索》*手册中的“关于搜索头群集化”。

索引器群集的各个部分

索引器群集是一组协同工作的 Splunk Enterprise 实例或**节点**，协同工作，提供冗余索引和搜索操作。每个群集有三种类型的节点：

- 单个**管理器节点**，用于管理群集。
- 几个到多个**对等节点**，用于维护数据的多个副本及为其创建索引，以及搜索数据。
- 一个或多个**搜索头**，用于协调对等节点集的搜索。

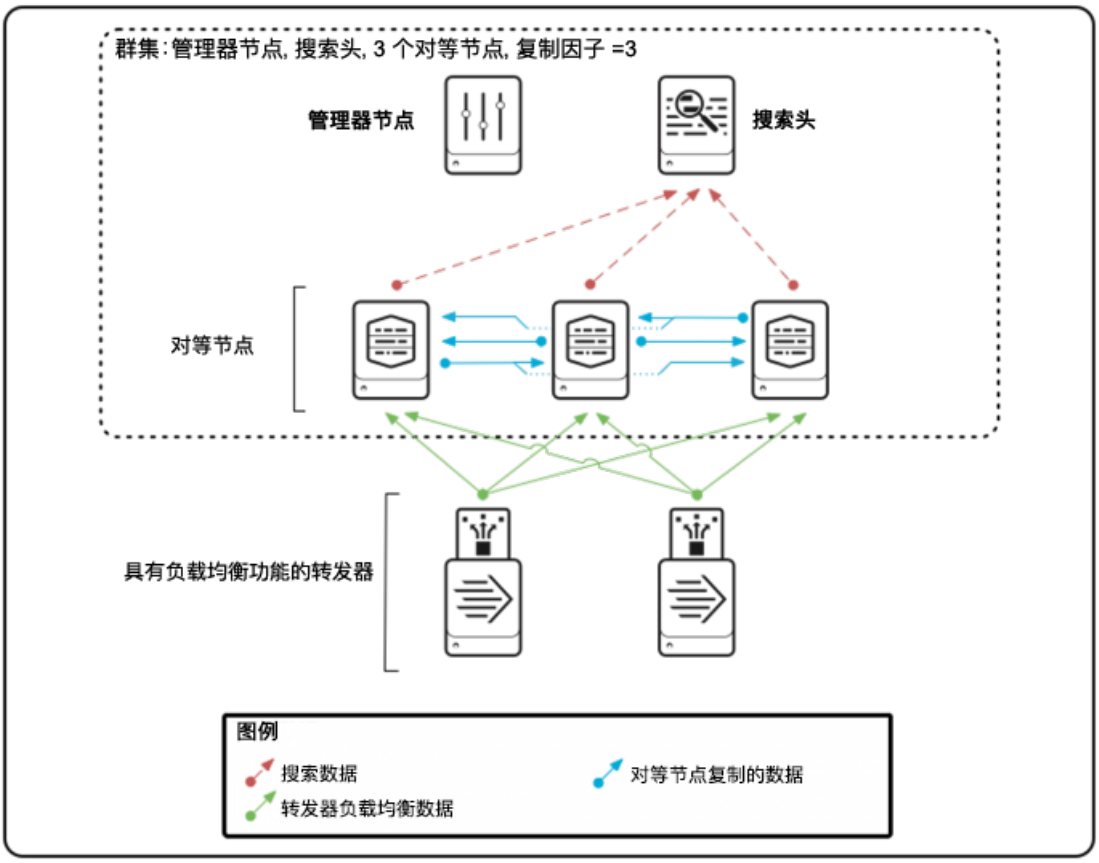
管理器节点会管理群集。它会协调对等节点之间的复制活动，并向搜索头指出查找数据的位置。它还会帮助管理对等节点的配置并在对等节点故障时安排补救活动。

对等节点将接收传入的数据并为其建立索引，就像非群集独立索引器一样。但与独立索引器不同的是，对等节点还会复制群集中其他节点的数据。对等节点可以为自己的传入数据建立索引，同时还将存储来自其他节点的数据副本。您的对等节点数必须至少与复制因子相同。也就是说，若要支持复制因子为 3，您需要至少三个对等节点。

搜索头将在对等节点集上运行搜索。您必须使用搜索头来管理索引器群集上的搜索。

大多数情况下，建议您使用**转发器**将数据传送到群集中。

以下是一个基本的、单站点索引器群集图，它包含三个对等节点并支持复制因子为 3：



上图是一个简单的部署，与一个小规模的非群集部署相似，其中包含了将负载均衡数据发送到一组索引器（对等节点）的一些转发器，以及将搜索结果发送到搜索头的索引器。有两个项目您在非群集部署中找不到：

- 索引器会将数据副本流送到其他索引器。
- 管理器节点，虽然它不参与任何数据流送，但会协调与搜索对等节点和搜索头相关的一系列活动。

多站点索引器群集

多站点索引器群集可帮您维护位于多个位置的索引数据的完整副本。这样便提供了加强灾难恢复和**搜索相关性**的优势。您可以指定每个站点上数据副本的数量。多站点群集在很多方面和基本的、单个站点群集类似，只在配置和行为上有一些不同。请参阅“多站点索引器群集”。

如何设置群集

群集的设置非常简单。设置群集的过程与设置一组独立索引器的过程相似。简单来说，就是安装索引器并执行一些配置操作。

主要区别在于，您还需要确定和启用群集节点。您要将一个 Splunk Enterprise 实例指定为管理器节点，将其他实例指定为对等节点。您的对等节点数必须至少与复制因子的大小相同。要为横向扩展增加索引容量，只需添加更多的对等节点即可。

您还需要设置一个或多个搜索头来管理对等节点上的搜索并合并用户的结果。

用与您在 Splunk Enterprise 中配置设置相同的方式启用群集节点：通过 Splunk Web 或 CLI，或直接编辑配置文件。

请参阅“部署索引器群集”。

如何搜索群集

您搜索群集的方式与搜索任何非群集索引器组的方式相同。您通过搜索头提交您的搜索。

但是后台的处理方式略有不同。提交搜索之后，搜索头将咨询管理器节点来确定当前的一组对等节点。然后，搜索头将搜索任务直接分发到这些对等节点。这些对等节点负责完成各自的任务并将结果发回到搜索头，搜索头将合并后的结果返回 Splunk Web。从用户的立场看，这与搜索任何独立索引器或非群集索引器组并没有任何不同。请参阅“在索引器群集中搜索如何工作”。

有了多站点群集，您还可以实现搜索相关性。在搜索相关性中，搜索头很可能从其站点本地的索引器获得搜索结果。同时，搜索仍然能访问完整的数据集合。请参阅“在多站点索引器群集中实现搜索相关性。”

预备知识

群集易于设置和使用，但您首先需要清楚地了解 Splunk Enterprise 索引和部署的基础知识。在执行任何操作前，请确保您了解以下内容：

- **如何配置索引器。**请参阅“索引器如何存储索引”，以及本手册中介绍管理索引的其他主题。
- **搜索头的作用。**有关分布式搜索和搜索头的介绍，请参阅《*分布式搜索*》手册中的“关于分布式搜索”。
- **如何使用转发器将数据导入到索引器。**请参阅《*数据导入*》手册中的“使用转发器”。

是否从非群集 Splunk Enterprise 部署迁移？

群集索引器有几个与非群集索引器不同的要求。在迁移索引器之前，知道这些问题很重要。有关详细信息，请参阅“群集和非群集索引器 Splunk Enterprise 部署之间的关键差异”。阅读完上述资料后，请转到“将非群集索引器迁移到群集环境”，了解有关实际迁移过程的详细信息。

多站点索引器群集

在 Splunk Enterprise 6.1 中，索引器群集有内置的站点识别，就是说，您可以以显式方式逐个配置**多站点索引器群集**。这样便简化并扩展了实施特定群集（如跨多个物理站点的群集，数据中心）的能力。

使用案例

多站点群集相对于单个站点群集来说，有两个关键优势：

- **改进的灾难恢复。**当灾难在一处发生时，通过在多处保存数据副本，您还可保留对数据的访问。多站点群集提供站点故障转移功能。如果一个站点有故障，建立索引和搜索可以在其余站点继续进行，而不会中断或丢失数据。
- **搜索相关性。**如果您配置每个站点使其都具有一个搜索头和全部可搜索数据，在每个站点的搜索头就能将其搜索限制在本地对等节点。这样在正常条件下就不需要搜索头去访问其他站点的数据，极大地减少了站点之间的网络流量。

多站点配置

配置多站点群集与配置基本单个站点群集相比有所不同。以下是多站点群集主要的不同点：

- 为每个节点分配一个站点。
- 按站点逐个指定复制和搜索因子。就是说，您可以指定想要保留在每个站点的副本数量和可搜索副本数量，以及您想要保留在群集中的总体数量。

还有一些其他的配置差异。请参阅“多站点部署概述”。

多站点架构

单个站点和多站点群集的架构相似。多站点群集的主要差异如下：

- 每个节点都属于一个分配的站点。
- 以站点识别的方式进行数据桶副本复制。
- 可行情况下，搜索头只将搜索分布到本地对等节点。

请阅读“多站点索引器群集架构”，获得更多关于多站点群集架构的信息。

相关信息

下面几个章节和主题详细介绍多站点群集：

- “部署和配置多站点索引器群集”。本章主题介绍多站点配置，包括搜索相关性配置、多站点复制和搜索因子。
- “管理多站点索引器群集”。本章介绍一系列问题，如：处理管理器站点故障，以及将多站点群集转换为单个站点。
- “将索引器群集从单个站点迁移到多站点”。本主题介绍如何将单个站点群集转换为多站点。

本手册中的其他主题在必要时会区分多站点和单个站点群集。

索引器群集架构的基础知识

本主题将介绍**索引器群集**架构。介绍单个站点群集的节点以及它们如何协同工作。此外，还会介绍一些基本概念，并概述群集对索引和搜索的处理方式。

多站点群集架构与单个站点群集架构相似。但是有些地方仍有显著差别。请阅读“多站点索引器群集架构”，以获得关于多站点群集架构及其与单个站点群集架构的区别的信息。

有关群集架构的更深入的讨论，请参阅“索引器群集如何工作”一章。

有关 SmartStore 索引的群集架构有何不同的信息，请参阅“SmartStore 架构概述”和“索引器群集操作和 SmartStore”。

群集节点

群集包括三种类型的节点：

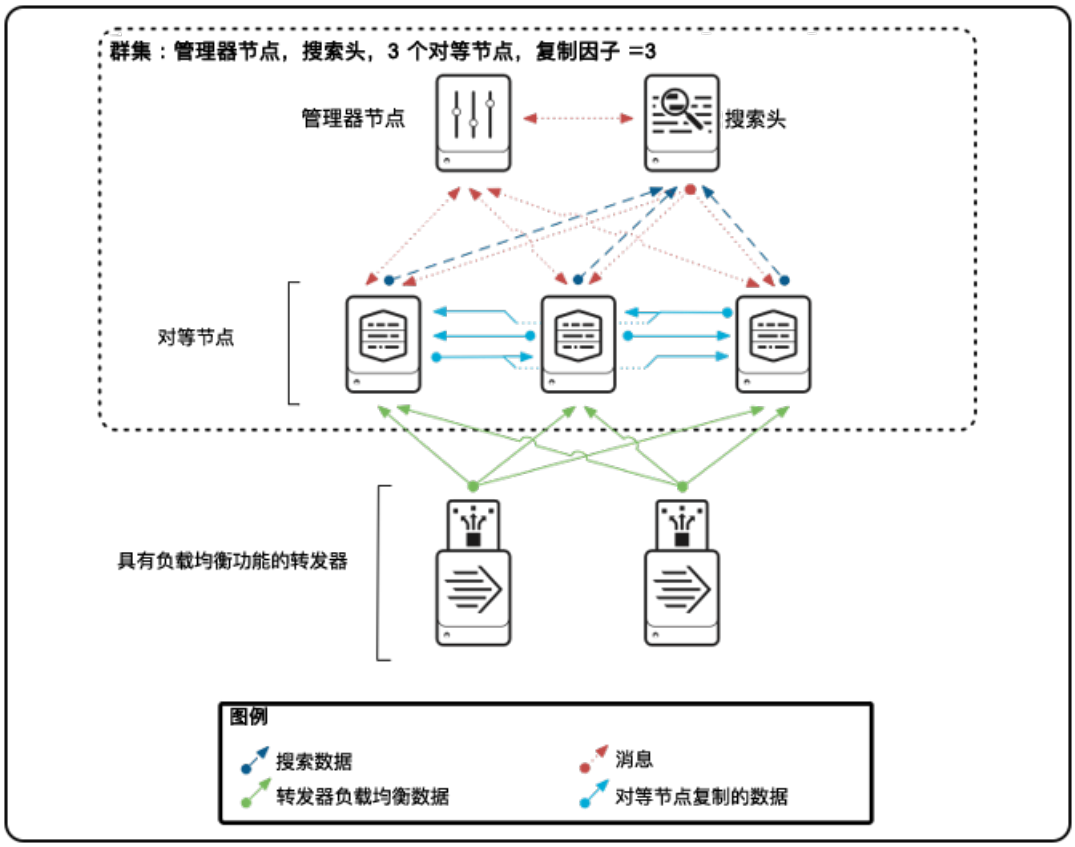
- 单个**管理器节点**，用于管理群集。
- 多个**对等节点**，用于索引和复制数据并对数据进行搜索。
- 一个或多个**搜索头**，用于协调所有对等节点的搜索。

此外，群集部署通常使用**转发器**来获取数据并将其转发到对等节点。

管理器节点、对等节点以及搜索头都是 Splunk Enterprise 的专用实例。每个节点必须驻留在单独的实例和单独的计算机上。例如，管理器节点不能像对等节点或搜索头一样驻留于同一个实例或同一台计算机上。

管理器节点和所有对等节点必须特定于单个群集。管理器节点不能管理多个群集。对等节点无法连接到多个管理器节点。但是，搜索头可以跨多个群集搜索。

以下是一个简单的单个站点群集图，其中包含若干对等节点以及若干向这些对等节点发送数据的转发器：



该图中发生的某些事件可能没有意义，请继续阅读下文。

管理器节点

管理器节点会管理群集。它会协调对等节点之间的复制活动，并向搜索头指出查找数据的位置。它还会帮助管理对等节点的配置并在对等节点脱机时安排补救活动。

与对等节点不同的是，管理器节点不为**外部数据**建立索引。一个群集只有一个管理器节点。

对等节点

对等节点用于执行群集的索引功能。它们将接收传入的数据并为其建立索引。此外，还负责将复制的数据发送到群集中的其他

对等节点，并接收来自其他对等节点的复制的数据。对等节点可以为自己的外部数据建立索引，同时接收和发送复制的数据。像所有索引器一样，对等节点还会搜索自己的索引数据，以响应搜索头的搜索请求。

部署的对等节点数量取决于两个因素：群集**复制因子**和索引负载。例如，如果复制因子为 3（这表示您想要存储数据的三个副本），则至少需要三个对等节点。如果您的索引负载超出三个索引器的处理能力，则可以添加更多的对等节点来增加容量。

搜索头

搜索头会管理对等节点集的搜索。它负责将搜索查询分布到对等节点并合并结果。您将从搜索头启动所有搜索。一个群集必须至少具有一个搜索头。

转发器

转发器的功能与任何 Splunk Enterprise 部署中的功能都相同。它们获取来自外部来源的数据，然后将此数据转发到索引器，对于群集而言，就是转发到对等节点。您并不需要使用转发器将数据导入到群集，但在大多数情况下，您都希望如此。这是因为，只有使用转发器，您才能启用**索引器确认**，通过这种方式来确保可靠地为传入数据建立索引。此外，为了处理可能出现的对等节点故障，建议使用**负载均衡**转发器。这样，如果一个对等节点发生故障，转发器可以将其转发切换到负载均衡组中的其他对等节点。有关群集环境中转发器的更多信息，请参阅本手册中的“使用转发器将数据导入索引器群集”。

重要概念

要了解群集的功能，需要熟悉几个概念：

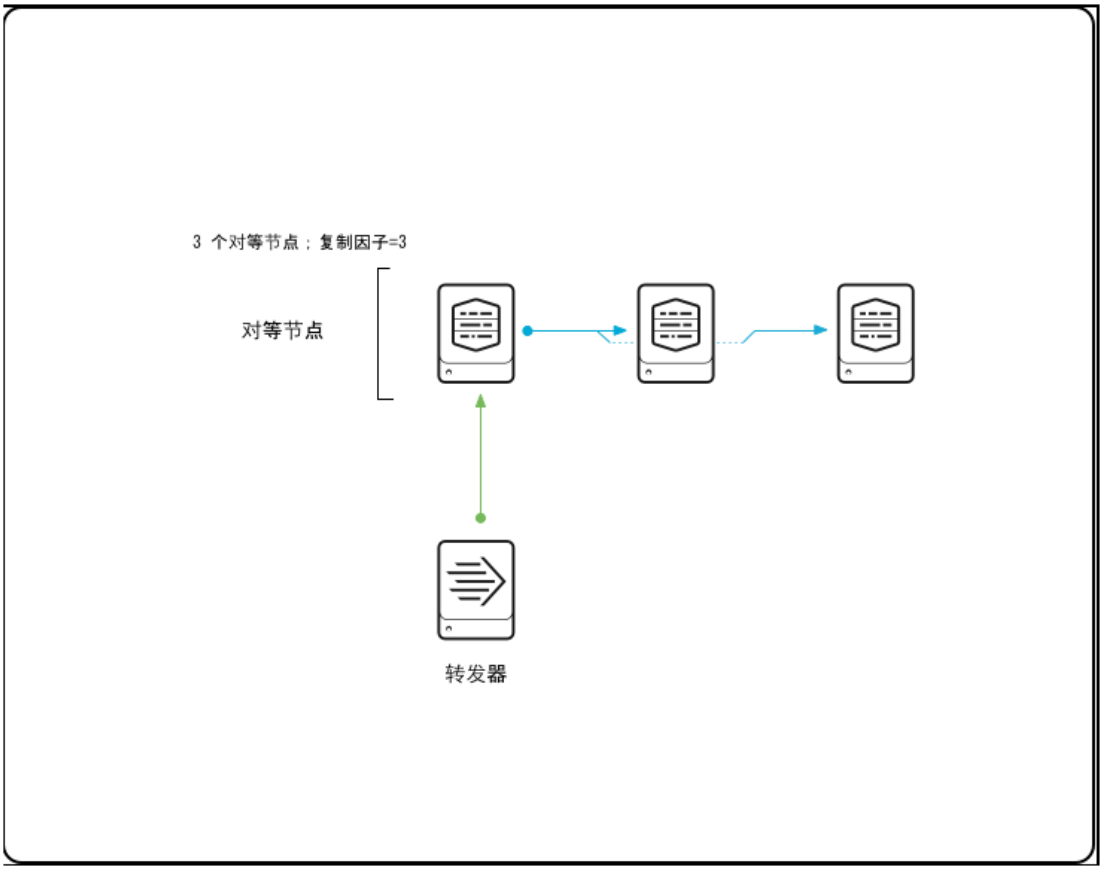
- **复制因子**。用于确定群集保留的数据副本数，因此，也确定了群集的故障容错基本级别。
- **搜索因子**。用于确定群集保留的可搜索的数据副本数，因此，也确定了在对等节点故障后群集恢复其搜索功能的速度。
- **数据桶**。数据桶是索引存储的基本单位。群集会保留与每个数据桶复制因子数相同的副本数。

本部分将简要介绍这些概念。

复制因子

在配置管理器节点时，您将指定群集要保留的数据副本数。此副本数称为群集的**复制因子**。复制因子是索引复制中的一个关键概念，因为它决定了群集的故障容错：群集能容许（复制因子 - 1）对等节点的故障。例如，要确保系统可以处理两个对等节点出现故障，必须将复制因子配置为 3，这意味着群集将三个相同的数据副本存储在单独的节点上。如果其中两个对等节点故障，第三个对等节点上的数据仍是可用的。复制因子的默认值是 3。

下图为某群集的高级表示，其中包含三个对等节点，复制因子为 3：



在该图中，一个对等节点将接收来自转发器的数据，它对此数据进行处理，然后将其流送到其他两个对等节点。群集中将包含对等节点数据的三个完整副本。该图以高度简化的方式表示对等节点复制，其中所有数据将通过单个对等节点进入系统中。在大多数由三个对等节点组成的群集中，所有三个对等节点都将从转发器接收外部数据，以及从其他对等节点接收复制的数据。

有关复制因子以及调整该值的考虑因素的详细讨论，请参阅“复制因子”主题。

重要提示：多站点群集使用完全不同的复制因子版本。请参阅“多站点复制和搜索因子”。

搜索因子

配置管理器节点时，指定**搜索因子**。搜索因子决定了群集保留的**可立即搜索**的数据副本的数量。

可搜索的数据副本比**不可搜索的**副本需要更多的存储空间，因此最好对搜索因子的大小加以限制，只要能够满足实际需要即可。在大多数情况下，使用默认值 2。这样在单个对等节点出现故障时，群集可以在几乎不受影响的情况下继续执行搜索。

对于某些数据而言，可搜索副本与不可搜索副本之间的差异如下：可搜索副本包含数据本身，以及群集用来搜索数据的一些广泛的索引文件。不可搜索副本只包含数据。即使数据存储在不可搜索副本中，但是已经执行了初步处理并采用了适当存储形式，以便在以后需要时可以重新创建索引文件。

有关搜索因子以及调整该值的考虑因素的详细讨论，请参阅“搜索因子”主题。

重要提示：多站点群集使用完全不同的搜索因子版本。请参阅“多站点复制和搜索因子”。

数据桶

Splunk Enterprise 将索引数据存储在**数据桶**（即包含数据文件的目录）中。一个索引通常由多个数据桶组成。

完整群集会保留与每个数据桶复制因子数相同的副本数，每个副本驻留在一个单独的对等节点上。数据桶副本有可搜索和不可搜索两种类型。完整群集会保留与每个数据桶搜索因子相同的可搜索副本数。

数据桶包含两种类型的文件：**原始数据文件**，该文件包含数据和一些元数据，并且对于可搜索的数据桶副本，还包含数据**索引文件**。

群集按数据桶复制数据。原始数据桶及其在其他对等节点上的副本包含相同的原始数据组。可搜索副本还包含索引文件。

每当某个对等节点创建新数据桶时，它会与管理器节点通信，获取可将数据桶数据流入的对等节点列表。如果您的群集中对等

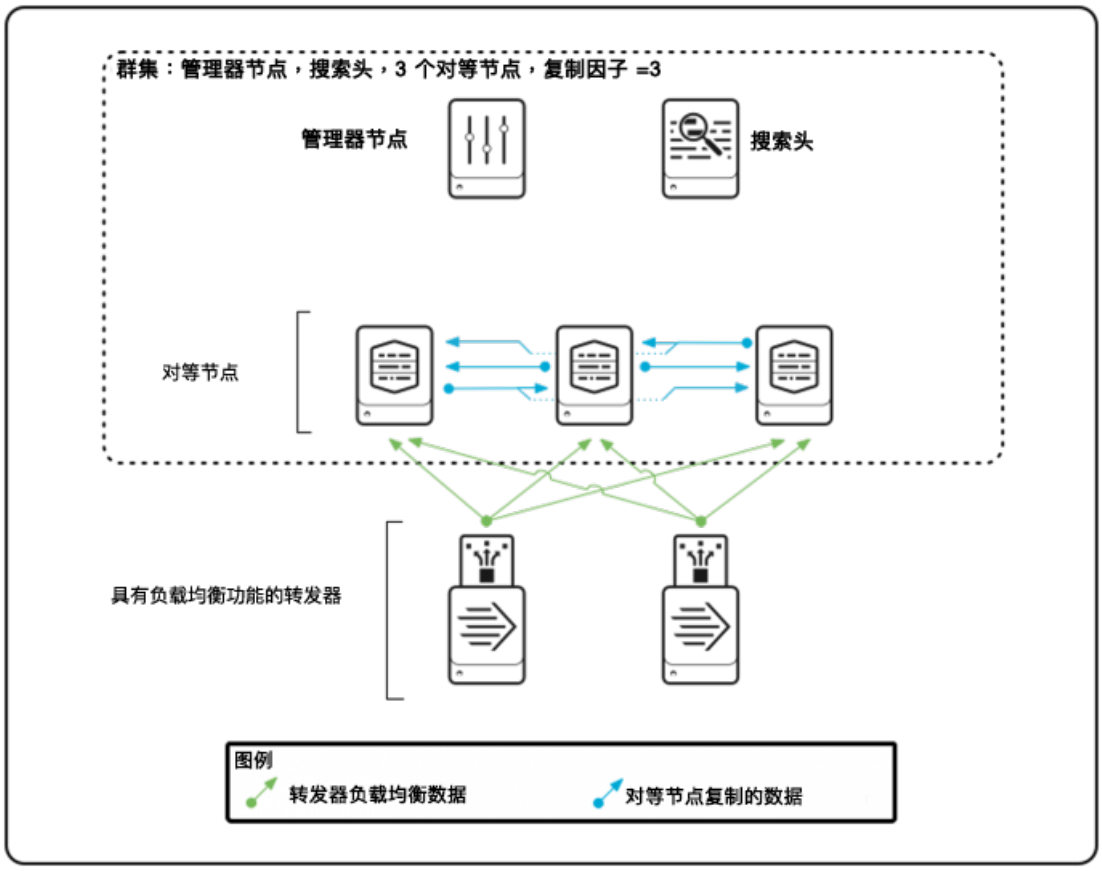
节点的数量大于复制因子，则每当某个对等节点创建新数据桶时，它可能会让其数据流入另外一组对等节点。最终，该对等节点原始数据桶的副本可能分布在很多个对等节点上，即使复制因子仅为 3。

您只有深刻了解数据桶才能了解群集架构。有关数据桶的一般概述，请参阅“索引器如何存储索引”，然后参阅“数据桶和索引器群集”主题。该主题会提供对于群集部署特别重要的数据桶概念的详细资讯。

索引如何工作

群集索引的工作方式类似于非群集索引，只不过群集存储数据的多个副本。每个对等节点接收、处理外部数据以及为其创建索引，行为与所有非群集索引器相同。主要不同之处在于，对等节点还会将已处理数据的副本流送或“复制”到群集中的其他对等节点，然后，这些对等节点将这些副本存储在其自己的数据桶中。某些接收已处理数据的对等节点可能还会为其建立索引。复制因子用于确定接收数据副本的对等节点数。搜索因子用于确定为数据建立索引的对等节点数。

对等节点可以为外部数据创建索引，同时存储从其他对等节点发送给自己的复制的数据的副本并可能为其创建索引。例如，如果您的群集中有三个对等群集，配置的复制因子为 3，每个对等节点都可以插入外部数据并为其创建索引，同时还存储其他对等节点所流入的复制的数据的副本。如果群集的搜索因子为 2，则接收流数据副本的其中一个对等节点还将为其建立索引。（此外，最初插入该数据的对等节点将始终为自己的副本建立索引。）下图显示了数据到对等节点的移动情况，同时包括了来自转发器以及其他对等节点的数据移动：



可将群集设置为所有对等节点均可插入外部数据。这是最常用的方案。只需在每个对等节点上配置输入便可实现此目的。不过，您还可以将群集设置为只有一部分对等节点能够插入数据。无论如何在群集中分散您的输入内容，所有对等节点都会或者可能会同时存储复制的数据。管理器节点按数据桶逐个确定哪些对等节点将获取复制的数据。您不能对其进行配置，除非是在多站点群集化中，您可以指定每个站点的对等节点组接收的数据副本的数量。

除了复制外部数据的索引，对等节点还复制其内部索引，如 `_audit` 和 `_internal` 等。

管理器节点将管理对等节点与对等节点之间的相互作用。最重要的是，它会通知每个对等节点，哪些对等节点将向其中流入数据。一旦管理器节点完成通信任务，对等节点便彼此交换数据而无需管理器节点的涉入，除非某个对等节点出现故障。管理器节点还将跟踪哪些对等节点包含可搜索数据，并确保可用的可搜索数据的副本数始终与搜索因子相等。如果某个对等节点出现故障，管理器节点将协调补救活动。

有关详细信息，请参阅“群集式索引如何工作”主题。

请参阅“多站点索引”，了解关于索引在多站点群集中的工作方式的相关信息。

有关如何将索引和 SmartStore 结合使用的信息，请参阅“索引如何在 SmartStore 中工作”。

搜索如何工作

在索引器群集中，搜索头用于协调所有搜索。该进程类似于**分布式搜索**在非群集式环境中的工作方式。主要区别在于，搜索头依赖管理器节点来通知它的搜索对等节点是谁。另外，还有多个不同进程已准备就绪，以确保搜索在每个数据桶的一个且仅一个副本上发生。

为确保每个数据桶只有一个副本参与搜索，需要将群集中每个数据桶的其中一个可搜索副本指定为**主要**副本。搜索仅在主要副本集合中进行。

主要副本集合会随时间变化，例如，响应对等节点出现故障这一事实。如果故障节点上的某些数据桶副本为主要副本，则会将这些数据桶的其他可搜索副本指定为主要副本作为替代。如果没有其他可搜索副本（因为群集的搜索因子为 1），必须将不可搜索副本设置为可搜索副本，然后才能将其指定为主要副本。

只要对等节点加入或重新加入群集，管理器节点会跨一组对等节点重新平衡主要性，以尝试改进搜索负载的分布。请参阅“重新平衡索引器群集的主要数据桶”。

管理器节点将跟踪所有对等节点上的所有数据桶副本，而对等节点本身已知自身数据桶副本的状态。由此一来，在响应搜索请求时，对等节点就会知道要对哪些数据桶副本进行搜索。

搜索头会定期从管理器节点获得一个活动搜索对等节点的列表。为处理搜索，搜索头随后会直接与这些对等节点通信，如同在任何分布式搜索中那样，将搜索请求和知识软件包发送到对等节点并合并从对等节点返回的搜索结果。

例如，假定某个群集有 3 个对等节点，共维护 20 个数据桶，需要对这些数据桶执行搜索以执行搜索头发出的特定搜索请求。这 20 个数据桶的主要副本可以分散在所有三个对等节点上，其中有 10 个主要副本位于第一个对等节点上，6 个位于第二个上，最后 4 个位于第三个上。每个对等节点都会收到搜索请求，然后由自身确定是否将其数据桶的特定副本视为主要副本，因此需要参加搜索。

有关详细信息，请阅读“在索引器群集中搜索如何工作”主题。

重要提示：关于搜索在多站点群集中的工作方式，有几点关键差异。例如，群集中的每个站点通常有一个完整的主要数据桶集合，这样搜索头才能完全搜索其站点本地的所有数据。请参阅“多站点搜索”，获得更多信息。

有关如何将搜索和 SmartStore 索引结合使用的信息，请参阅“搜索如何在 SmartStore 中工作”。

群集如何处理对等节点故障

如果对等节点出现故障，管理器节点将进行协调，尝试在其他对等节点上重新生成该对等节点的数据桶。例如，如果故障节点存储 20 个数据桶副本，其中 10 个是可搜索副本（包括三个主要数据桶副本），管理器节点会直接在其他节点上创建这 20 个数据桶副本。它也将尝试使用其他节点上相同数据桶的可搜索副本替代这 10 个可搜索副本。同时，它会将其他对等节点上的相应可搜索副本的状态从非主要更改为主要以替换主要副本。

复制因子和节点故障

如果剩余的对等节点数小于复制因子所指定的数字，群集将无法更换这 20 个丢失的副本。例如，如果您的三节点群集的复制因子为 3，群集将无法在节点发生故障时替换丢失的副本，因为替代副本没有可以转到的其他节点。

然而，除非在极端情况下，群集应能够通过将其他对等节点上的这些数据桶的可搜索副本指定为“主要”副本来替换丢失的主要数据桶副本，以便搜索头可对所有数据继续进行完全访问。

只有在节点的复制因子数量下降时，群集才会无法保留一组完整的主要副本。例如，如果群集有五个对等节点，复制因子为 3，那么在一个或两个对等节点故障但第三个对等节点仍正常运行时，群集仍然拥有一组完整的主要副本集合。

搜索因子和节点故障

搜索因子决定在节点发生故障后是否可以迅速恢复完整搜索功能。为确保从一个故障节点快速恢复，搜索因子必须至少设置为 2。这让管理器节点可以使用其他节点上的现有可搜索副本立即替代出现故障节点上的管理器节点。

如果搜索因子设置为 10，这意味着群集只保留一个可搜索数据桶副本集合。如果带一些主要副本的对等节点发生故障，群集必须首先将剩余对等节点上的一组响应不可搜索副本转换为可搜索副本，然后才能将其指定为主要副本来替换丢失的主要副本。虽然执行了这一相当耗时的过程，但是群集拥有的仍是不完整的主要数据桶集合。搜索可以继续，但是仅跨可用主要数据桶。最终，群集将更换所有缺失的主要副本。之后，即可对完整的数据集执行搜索。

在另一方面，如果搜索因子至少为 2，群集可以立即分配主要状态给剩余节点上的可搜索副本。替换故障节点上的可搜索副本的活动仍会发生，但在此期间，可以不受干扰地继续对所有群集数据执行搜索。

有关对等节点故障的详细信息，请参阅“对等节点故障时的情况”。

请参阅“多站点索引器群集如何处理对等节点故障”，了解多站点群集处理对等节点故障的相关信息。

群集如何处理管理器节点故障

如果管理器节点发生故障，在一段时间内，对等节点会继续为数据创建索引及复制数据，搜索头会继续在数据中搜索。但最终仍会导致问题出现，尤其是其中一个对等节点故障后。没有管理器节点，将无法从对等节点丢失状况恢复，之后，搜索头将在一组不完整的数据中搜索。总之，没有管理器节点，群集将继续以它所能做到的最佳状态运行，但系统处于不一致状态，且结果无法得到保证。

有关管理器节点故障的详细信息，请阅读“管理器节点故障时的情况”。

多站点索引器群集架构

本主题介绍**多站点索引器群集**的架构。主要侧重于多站点群集与单个站点群集的差异。请参阅“索引器群集架构的基础知识”，获得群集架构的概览（侧重于单个站点群集）。

多站点与单个站点架构的差异

多站点群集与单个站点群集主要有以下方面的差异：

- 每个节点（管理器节点/对等节点/搜索头）有一个分配的站点。
- 站点识别时会有数据桶副本的复制。
- 只要有可能，搜索头只将搜索分布到本地对等节点。
- 适用情况下，数据桶修复活动应遵守站点界限。

多站点群集节点

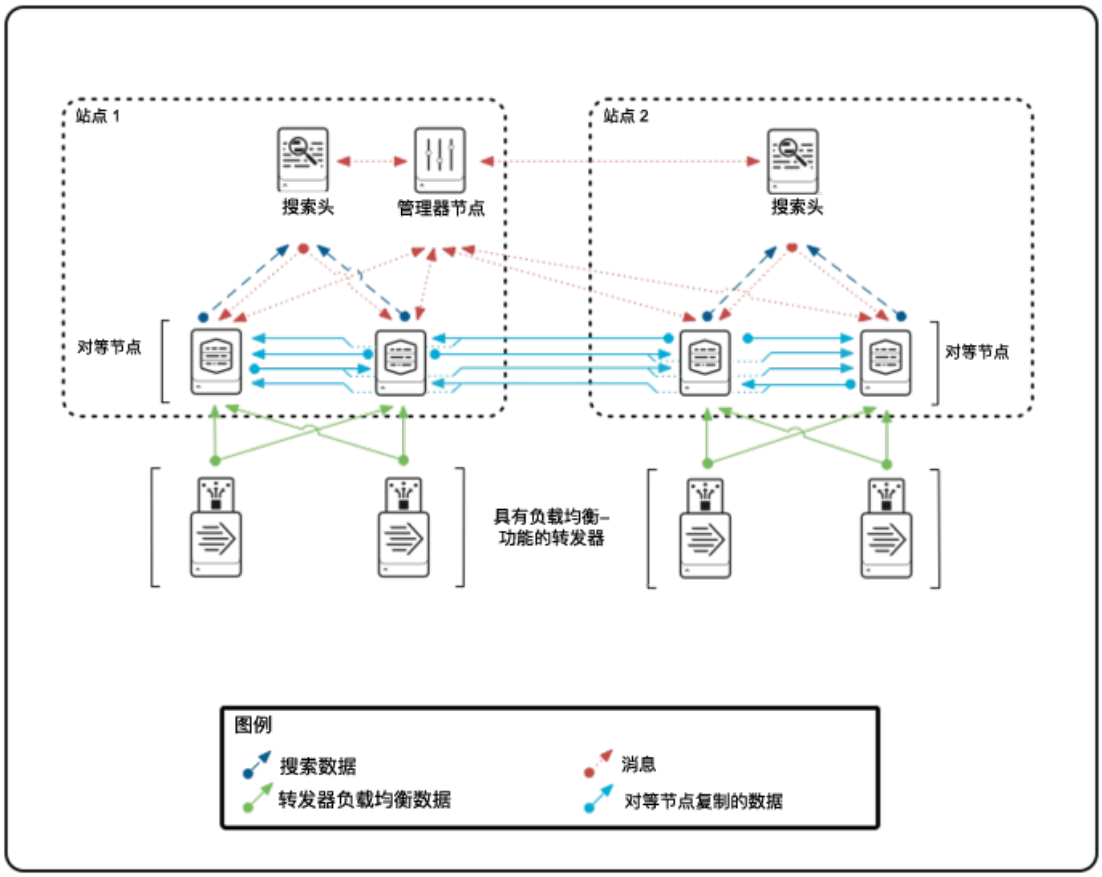
多站点和单个站点节点有如下共同特性：

- 群集有三种类型的节点：管理器节点、对等节点，以及搜索头。
- 每个群集只有一个管理器节点。
- 群集的对等节点和搜索头的数量不受限制。

多站点节点有如下不同点：

- 每个节点属于一个特定的站点。通常，物理位置决定一个站点。就是说，如果您想要群集涵盖波士顿和费城的所有服务器，则需要将所有波士顿的节点分配给 site1，将所有费城的节点分配给 site2。
- 通常，多站点群集在每个站点都有搜索头。这对于**搜索相关性**十分必要，搜索相关性通过允许搜索头从本地访问所有数据来提高搜索效率。

下面是一个双站点群集示例。



请注意以下事项：

- 管理器节点控制整个群集。尽管管理器节点在物理上位于一个站点，管理器节点实际上却不属于任何一个站点。但是，每个管理器节点有一个内置搜索头，该搜索头会要求您整体为管理器节点指定一个站点。注意，管理器节点的搜索头仅用于测试目的。请勿将其用于生产环境。
- 这是一个配置为搜索相关性的群集示例。每个站点有它自己的搜索头，可在其站点搜索对等节点集。但是，搜索头可能也会搜索本站点之外的对等节点，取决于具体情况。请参阅“多站点搜索和搜索相关性”。
- 节点跨站点界限复制数据。此行为对于灾难恢复和搜索相关性都非常关键。

多站点复制和搜索因子

与单个站点对应情况相同，多站点复制和搜索因子分别决定群集中的数据桶副本和可搜索数据桶副本的数量。区别是多站点复制和搜索因子也决定每个站点上的副本数量。一个三站点群集的多站点复制因子可能如下所示：

```
site_replication_factor = origin:2, site1:1, site2:1, site3:1, total:4
```

此复制因子指定每个站点将获取每个数据桶的一份副本，除非站点是数据的源站点（此种情况它将获得两份副本）。它还指定整个群集中副本总量为四个。

在此特定示例中，复制因子显式指定所有站点，但这并非必须操作。**显式站点**是复制因子进行显式指定的站点。**非显式站点**是那些复制因子不进行显式指定的站点。

这是三站点群集的多站点复制因子的另一个示例。此复制因子仅明确指定两个站点：

```
site_replication_factor = origin:2, site1:1, site2:2, total:5
```

在此示例中，非显式站点 `site3` 获取的副本数量不同。如果 `site1` 是源站点，`site1` 获得两份副本，`site2` 获得两份副本，`site3` 获得剩余的一份副本。如果 `site2` 是源站点，`site1` 获得一份副本，`site2` 获得两份副本，`site3` 获得两份副本。如果 `site3` 是源站点，`site1` 获得一份副本，`site2` 获得两份副本，`site3` 获得两份副本。

注意：在本例中，`total` 值不能为 4。必须至少为 5。这是因为，当复制因子有非显式站点时，副本总数必须至少是所有显式站点和原始值的和。

关于复制因子的语法和行为的详细内容，请参阅“配置站点复制因子”。

站点搜索因子的工作方式也一样。详细内容，请参阅“配置站点搜索因子”。

多站点索引

多站点索引和单个站点索引类似，在“索引器群集架构的基础知识”中有介绍。单个管理器节点协调所有站点上的所有节点间的复制。

本部分通过复制因子为下列内容的三站点群集示例，简要介绍多站点的主要差异：

```
site_replication_factor = origin:2, site1:1, site2:1, site3:1, total:4
```

需要注意的多站点问题主要有：

- 数据复制基于复制因子，不受限于站点界限。在本例中，如果 site1 中的节点获取数据，则它将数据的一份副本流出给 site1 中的另一个节点（以满足 origin 设置为 2），一份副本发送给 site2 中的一个节点，一份副本发送给 site3 中的一个节点。
- 多站点复制含有源站点的概念，它允许群集为生成数据的站点以不同方式处理数据。此示例说明了这一点。如果 site1 生成数据，则它获得两份副本。如果另一站点生成数据，则 site1 仅获得一份副本。
- 与单个站点复制相同，您不能指定确切的节点去接收复制的数据。然而，您可以指定其对等节点接收数据的站点。

关于群集如何处理迁移的单个站点数据桶的信息，请参阅“从单个站点迁移索引器群集到多站点”。

多站点搜索与搜索相关性

多站点搜索在很多方式上与单个站点搜索类似，“索引器群集架构的基础知识”中有介绍。每个搜索都在一组主要数据桶副本上运行。然而，有一个关键差异。

多站点群集提供**搜索相关性**，搜索相关性允许搜索在站点本地数据上运行。您必须配置群集来利用搜索相关性。具体地说，您必须确保可搜索数据和搜索头都能在本地获得。

要达到此目的，可配置搜索因子，这样每个站点至少有一组完整的可搜索数据。然后，只要此站点工作正常，管理器节点即保证每个站点有一组完整的主要数据桶副本。这就是所谓的**有效**状态。

借助搜索相关性，搜索头仍然会将其搜索请求分布到群集中的所有节点，但是仅有与搜索头位于同一站点的节点会通过搜索其主要数据桶副本并将结果返回到搜索头来响应请求。

如果一些站点的节点丢失意味着它不再有一组完整的主要副本（这样就不再是有效状态），数据桶修复行为会尝试将站点恢复到有效状态。在修复期间，远程站点上的节点将根据需要参与搜索，来保证搜索头仍然能获得一组完整的结果。在站点重新获得其有效状态后，搜索头再次仅使用本地节点来完成其搜索。

注意：如果需要，您可以禁用搜索相关性。当对于特定的搜索头禁用搜索相关性时，该搜索头可以从任意站点上的对等节点获取它的数据。

关于搜索相关性以及如何配置搜索因子来支持搜索相关性的更多信息，请参阅“在多站点索引器群集中实现搜索相关性”。关于内部搜索过程（包括搜索相关性）的更多信息，请参阅“索引器群集搜索如何工作”。

多站点群集和节点故障

多站点群集处理节点故障的方法与单个站点群集有一些显著差别。

阅读本部分之前，您必须理解“保留”数据桶副本的概念。

保留的数据桶副本是一个正等着最终分配给一个对等节点的虚拟副本。当副本处于保留状态时，它实际上尚未真正存在于存储中。一旦管理器节点将其分配给可用的对等节点，数据桶就会以通常的方式流送到该对等节点。

作为多站点对等节点故障导致的后果，一些数据桶副本可能不会被立即分配给一个对等节点。例如，在一个总复制因子为 5 的群集中，管理器节点可能通知原始对等节点将数据桶流送到其他三个对等节点。这导致四个副本（原始的加上三个流送副本）与第五个副本一起等着在一定条件满足时分配给一个对等节点。第五个未分配的副本称为保留副本。本部分说明了当对等节点故障时群集必须以怎样的方式保留副本。

多站点群集如何处理对等节点故障

当一个节点有故障时，如有可能，同一站点会执行数据桶修复操作。群集会尝试向该站点剩下的节点上添加副本来替换缺失的数据桶副本。（在所有情况下，每个节点最多拥有任何特定数据桶的一份副本。）如果在该站点内不能通过添加副本到节点来修复所有数据桶，那么，根据复制因子和搜索因子的情况，群集可能会从其他站点的节点上复制副本。

在这些情况下修复行为部分取决于故障节点是位于显式站点还是非显式站点。

如果一个显式站点上有太多的节点故障，以至于站点不能满足它自己站点特定的复制因子，则群集不会通过复制其他站点上节点的副本来补偿。它会假定所需数量的节点最终将回到该站点。同样，对于新数据桶，它为回到站点的节点预留副本。换句话说，它不分配那些副本给不同站点的对等节点，而是等待直到第一个站点有可用的对等节点然后分配副本给那些对等节点。

例如，为三站点群集（site1，site2，site3）提供这样的复制因子：

```
site_replication_factor = origin:2, site1:1, site2:2, total:5
```

群集通常在 site2 上保留两份副本。但是如果 site2 上有太多节点故障，以至于仅剩一个，站点不再满足其复制因子 2，则剩下的节点获得群集中所有数据桶的一份副本，群集为站点保留另一组副本。当第二个对等节点重新加入 site2，群集会将预留的副本流送到该节点。

当一个非显式站点丢失了太多节点以至于它不能再保持原有数量的数据桶副本时，则群集会通过添加副本到其他站点来弥补差异。例如，假定上例中的非显式 site3 有一些数据桶的两份副本，然后它丢失了所有，只剩下一个节点，所以它只能保留每个数据桶的一份副本。群集通过向其他站点的一个节点流送该数据桶的一份副本来补偿，是基于这样的假定：存在至少一个节点，这个节点还没有一份该数据桶的副本。

关于群集如何处理一个站点上所有对等节点都有故障的情况的详细信息，请参阅“群集如何处理站点故障”。

群集如何处理站点故障

站点故障仅是对等节点故障的一种特殊情况。群集修复的发生遵循先前对等节点故障中介绍的规则。特别需要注意的是，群集可能保留副本，针对最终站点的返回作预留。

对于现有数据桶的任何预留副本，群集在其修复活动期间不会添加副本到其他站点。类似的，对于在站点故障后添加的新数据桶，群集会保留一定数量的副本，直到站点返回到群集。

下面介绍群集如何决定预留的副本数量：

- 对于显式站点，群集预留副本和可搜索副本的数量为站点的搜索和复制因子所指定。
- 对于非显式站点，如果站点的搜索和复制因子的 total 组件足够大，则群集处理所有显式站点后预留一份可搜索副本，以容纳此副本。（如果搜索因子不够大，但是复制因子足够大，则群集预留一份不可搜索副本。）

例如，您有一个带两个显式站点（site1 和 site2）以及一个非显式站点（site3）的三站点群集，配置如下：

```
site_replication_factor = origin:2, site1:1, site2:2, total:5
site_search_factor = origin:1, site1:1, site2:1, total:2
```

在一个站点故障的情况下，群集会这样预留数据桶副本：

- 如果 site1 故障，群集会预留一份可搜索副本。
- 如果 site2 故障，群集会预留两份副本，其中包括一份可搜索副本。
- 如果 site3 故障，群集会预留一份不可搜索副本。

在修复现有数据桶期间或者在添加新数据桶期间，一旦预留的副本已定，群集会复制所有剩余的副本到其他可用站点。

当站点返回群集时，数据桶以必要的程度为该站点进行修复，以确保该站点保有至少其分配的预留的数据桶副本，包括新数据桶和站点发生故障时所有的数据桶。

如果发生故障的站点是管理器节点所在站点，您可启动剩下的一个站点上的备用管理器节点。请参阅“处理索引器群集管理器站点故障”。

多站点群集如何处理管理器节点故障

多站点群集处理管理器节点故障和单个站点群集一样。这种情况下，群集将继续发挥最佳作用。请参阅“管理器节点故障时的情况”。

部署索引器群集

索引器群集部署概述

本主题介绍了部署**索引器群集**的主要步骤。后续主题将对这些步骤加以详细介绍。

在尝试部署群集之前，必须先熟悉 Splunk Enterprise 管理的以下几个方面：

- **如何配置索引器。**具体信息请参阅“索引器如何存储索引”，以及本手册中介绍管理索引的其他主题。
- **搜索头的作用。**有关分布式搜索的介绍，请参阅《*分布式搜索*》手册中的“关于分布式搜索”。但是，请注意，配置索引器群集搜索头和其他搜索头有点不同。要获得关于差异的信息，请参阅本手册中的“搜索头配置概述”。
- **如何使用转发器将数据导入到索引器。**请参阅《*数据导入*》手册中的“使用转发器”。

重要提示：本章假定您正在索引器群集中部署独立搜索头。有关如何整合作为**搜索头群集**成员的搜索头的信息，请参阅《*分布式搜索*》手册中的“通过索引器群集集成搜索头群集”。

是否从非群集 Splunk Enterprise 部署迁移？

群集索引器（对等节点）有几个与非群集索引器不同的要求。在迁移索引器之前，知道这些问题很重要。请参阅“群集和非群集索引器部署之间的关键差异”。阅读完上述资料后，请转到“将非群集索引器迁移到群集环境”，了解有关实际迁移过程。

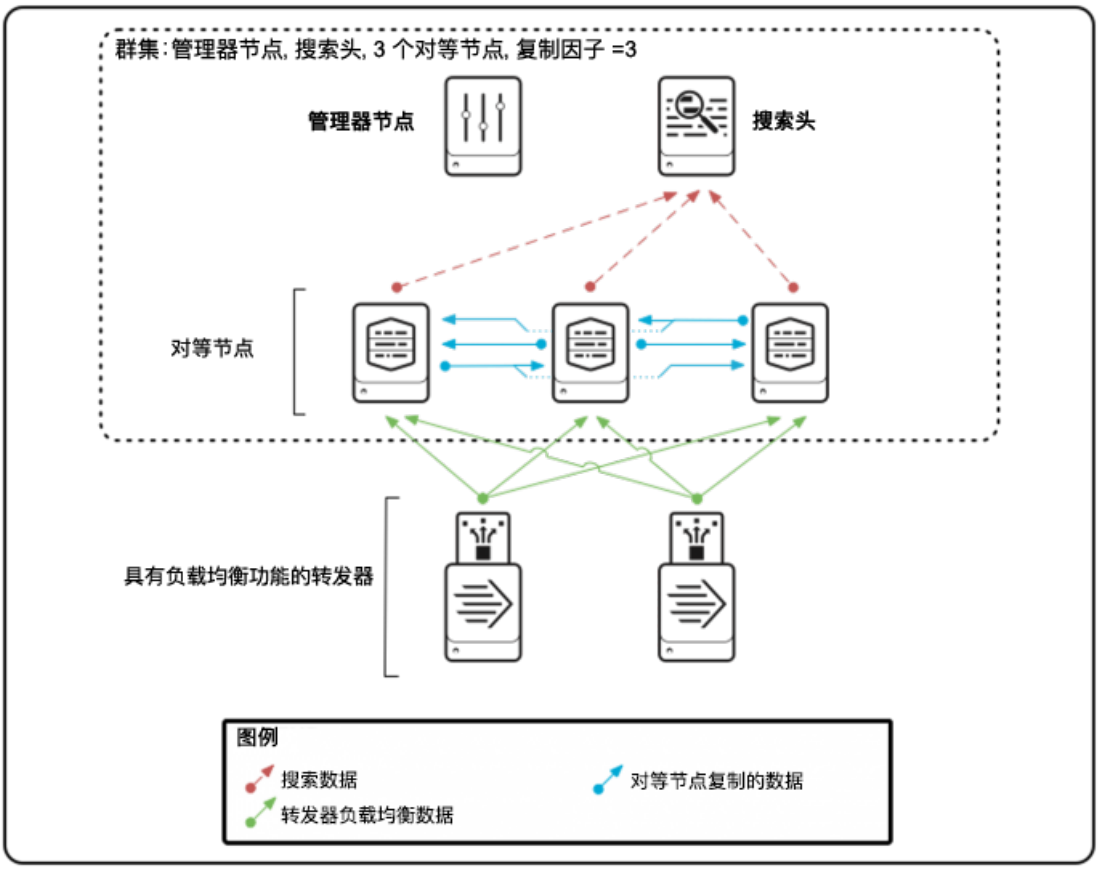
重要提示：在将一个索引器从非群集迁移到群集之前，请确认您的需求。此过程只能单向进行。不支持索引器从群集转换成非群集的过程。

部署一个多站点群集

与单个站点群集相比，**多站点索引器群集**相当复杂。部署它们需要考虑更多的问题，并执行一组完全不同的配置。如果您正在配置一个多站点群集，请先阅读本主题，然后阅读“多站点索引器群集部署概述”。

部署群集

部署群集时，可启用并配置群集管理器节点和群集对等节点来执行索引。还可启用搜索头来搜索群集中的数据。此外，您通常会设置转发器向群集发送数据。下面是一个小型群集图，其中显示您部署的各种节点：



以下是部署群集的主要步骤：

1. 确定您的要求：

a. 了解数据可用性和故障转移需求。请参阅“关于索引器群集”。

b. 确定部署基本的单个站点群集还是部署多站点群集。多站点群集具有强大的灾难恢复能力，因为它们允许您将数据副本分布在多个位置。它们还启用搜索相关性，这样就通过将搜索限制在本地数据，从而降低网络流量。有关更多信息，请参阅“多站点索引器群集”。

c. 决定要实现的**复制因子**。复制因子是指群集保留的原始数据副本的数量。优化复制因子取决于环境的特定因素，但实际上涉及故障容错与存储容量的权衡。较高的复制因子表示更多的数据副本将驻留在更多的节点上，因此您的群集可以容许更多的节点故障，而不会损失数据可用性。但还表示您将需要更多的节点和存储来处理其他数据。对于多站点群集，您还需要决定在每个站点上放多少份副本。有关更多信息，请参阅“复制因子”。

警告：确保先根据需要选择合适的复制因子。不建议在群集包含大量数据后增加复制因子。群集将需要执行大量数据桶复制，以与增加的复制因子相对应，这会大大降低复制时群集的总体性能。

d. 决定要实现的**搜索因子**。搜索因子指示群集保留多少可搜索的索引数据副本。这有助于确定群集从故障的节点中恢复的速度。较高的搜索因子可以让群集更快速地恢复，但同时还需要更多的存储空间和处理能力。对于大多数单个站点的部署，默认的搜索因子值 2 是一个合适的权衡，当节点故障时搜索通常仍能继续进行，几乎没有中断。对于多站点群集，您还需要决定在每个站点上放多少份可搜索副本。有关更多信息，请参阅“搜索因子”。

警告：务必先根据需要选择合适的搜索因子。不建议在群集包含大量数据后增加搜索因子。群集将需要执行大量处理（将不可搜索的数据桶副本转换成可搜索副本），以与增加的搜索因子相对应，这在处理时会对群集的总体性能产生不利的影响。

e. 确定同时决定群集大小的其他因素；例如，将要建立索引的数据量。通常，最好将所有索引器保留在单个群集中，因此要实现横向扩展，除了复制因子所需的节点之外，您还需要添加对等节点。同样，视预计的搜索负载而定，您可能需要增加多个搜索头。

f. 学习“索引器群集的系统要求和其他部署注意事项”主题，以了解有关其他关键问题的信息。

2. 在您的网络上安装 Splunk Enterprise 群集实例。至少需要（复制因子 + 2）个实例：

- 至少需要个数等于复制因子的**对等节点**，但可能需要添加更多的对等节点来提高索引容量，如步骤 1e 中所述。
- 还需要两个实例，一个用于**管理器节点**，另一个用于**搜索头**。

对于多站点群集，您必须考虑每个站点的搜索头和对等节点的需求，它们取决于搜索相关性和灾难恢复需求。请参阅“多站点索引器群集部署概述”。

有关如何安装 Splunk Enterprise 的信息，请阅读《*安装手册*》。

3. 在实例上启用群集化：

a. 启用管理器节点。请参阅“启用管理器节点”。

b. 启用对等节点。请参阅“启用对等节点”。

c. 启用搜索头。请参阅“启用搜索头”。

重要提示：对于多站点群集，启用群集节点的过程有所不同。请参阅“多站点索引器群集部署概述”。

4. 完成对等节点的配置：

a. 配置对等节点的索引设置。只有当您需要增添默认索引和应用时，才需要执行此步骤。通常情况下，所有对等节点必须使用相同的索引集，因此，如果您向一个对等节点添加索引（或定义索引的应用），则必须使用群集特定分发方法将它们添加到所有对等节点。可能还需要协调这组对等节点的其他配置。相关操作信息，请参阅“准备对等节点进行索引复制”。

b. 配置对等节点的数据导入。大多数情况下，最好使用转发器将数据发送到对等节点，如“将数据导入索引器群集的方式”中所述。如该主题中所述，您通常希望部署已启用索引器确认的负载均衡转发器。

启用了对等节点并为对等节点设置数据导入后，群集就会自动开始建立数据索引并复制数据。

5. 将管理器节点配置为将数据转发到对等节点。这个最佳做法提供了几个优势。请参阅“最佳做法：将管理器节点数据转发到索引器层”。

其他部署方案

本手册还为几个其他群集部署方案提供了指南：

- 向群集添加已具有数据的索引器。请参阅“将非群集索引器迁移到群集环境”。

- 在新的索引器群集上部署 **SmartStore** 索引。请参阅“在新的索引器群集上部署 SmartStore”。
- 将索引器群集上的现有数据迁移到 SmartStore。请参阅“将索引器群集上的现有数据迁移到 SmartStore”。
- 在索引器群集上启动 SmartStore 索引。请参阅“启动 SmartStore 索引”。
- 将单个站点群集迁移到多站点。请参阅“将索引器群集从单个站点迁移到多站点”。
- 使用群集完全是为了实现索引可调整性，此时不要求索引复制。请参阅“使用索引器群集调整索引”。

群集和非群集索引器部署之间的关键差异

本主题说明群集和非群集索引器之间的关键差异。特别讨论了有关系统要求和部署的问题。

如果计划将当前的一组索引器迁移至群集，请仔细阅读此主题。

不要将部署服务器或第三方部署工具与群集对等节点一起使用

不支持使用部署服务器或任何第三方部署工具（如 Puppet 或 CFEngine 等）作为向群集对等节点分发配置或应用的方法（索引器）。

要跨一组群集对等节点分布配置，请使用“更新通用对等节点配置”主题中概述的**配置软件包**方法。正如该主题介绍的那样，配置软件包方法涉及首先将对等节点应用放到管理器节点上，然后以协调的方式分发这些应用到对等节点。

有关如何将应用分布从部署服务器迁移到配置软件包方法的信息，请参阅“将应用迁移到群集”。

注意：您可以使用部署服务器分发更新给索引器群集中的搜索头，只要它们是独立搜索头。不能使用部署服务器分发更新给**搜索头群集**成员。

系统要求差别

与非群集索引器相比，对等节点有一些不同的系统要求。在迁移索引器之前，请阅读“索引器群集的系统要求和其他部署注意事项”主题。特别要注意以下差异：

- 将索引器转换为群集对等节点时，磁盘使用量会大幅上升。确保相对于每日索引量、搜索因子和复制因子，可用的磁盘空间足够大。有关对等节点磁盘使用情况的详细信息，请参阅“存储注意事项”。
- 群集节点不能共享 Splunk Enterprise 实例。管理器节点、对等节点和搜索头必须各自在自己的实例上运行。

其他注意事项和与非群集部署的差异

另外，还需注意以下事项：

- 对于大多数群集部署类型，应为向对等节点发送数据的转发器启用**索引器确认**。请参阅“索引器确认如何工作”。
- 您可以通过使用**索引器发现**功能来简化转发器与对等节点的连接过程。请参阅“索引器发现方法的优势”。
- 由于以下几个因素，性能总体上会有些降低。主要因素是索引器确认。另外，对来自其他对等节点的复制的数据进行存储和可能建立索引的需要也对性能有些影响。
- 重新启动群集对等节点时，应使用 Splunk Web 或一个可识别群集的 CLI 命令，如 `splunk offline` 或 `splunk rolling-restart`。不要使用 `splunk restart`。有关详细信息，请参阅“重新启动整个索引器群集或单个群集节点”。

迁移非群集索引器

要了解如何将现有索引器迁移到群集以及这样做的后果，请阅读主题“将非群集索引器迁移到群集环境”。

索引器群集的系统要求和其他部署注意事项

索引器群集是指 Splunk Enterprise 索引器组，因此一般情况下，您只需遵守索引器的系统要求。有关索引器的详细软件和硬件要求，请阅读*《安装手册》*中的“系统要求”。当前主题指出了群集的其他要求。

注意：对于具有 **SmartStore** 索引的群集，系统要求会有些不同。请参阅“SmartStore 系统要求”。

关键要求摘要

以下是需要注意的主要问题：

- 每个群集节点（管理器、对等或搜索头）必须驻留在单独的 Splunk Enterprise 实例上。
- 每个节点实例必须运行在单独的计算机或虚拟机上，同时每台计算机必须运行相同的操作系统及版本。
- 所有节点必须连接到网络。
- 各群集节点之间有严格的版本兼容性要求。

例如，要部署由三个对等节点、一个管理器节点和一个搜索头组成的群集，您需要五个 Splunk Enterprise 实例在有网络连接

的五台计算机上运行。所有计算机必须运行相同的操作系统及版本。

这里还需要注意一些其他问题：

- 与非群集部署相比，群集需要更多存储空间以容纳多个数据副本。
- 索引复制本身不增加您的许可需求。
- 您无法使用部署服务器分发更新到对等节点。

有关详细信息，请参阅本主题的剩余部分。

需要的 Splunk Enterprise 实例

每个群集节点必须驻留在自身的 Splunk Enterprise 实例中。因此，群集必须至少包含（复制因子 + 2）个实例：对等节点最小**复制因子**数，加上一个管理器节点和一个或多个搜索头。例如，如果您想要以复制因子 3 部署一个群集，则必须至少设置五个实例：三个对等节点、一个管理器节点，以及一个搜索头。要了解有关复制因子的更多信息，请参阅本手册中的“复制因子”。

群集大小除复制因子之外还取决于其他因素，如需要建立索引的数据量。请参阅“索引器群集部署概述”。

尽管管理器节点具有搜索功能，但是您应仅使用这些功能用于调试目的。管理器节点的资源必须专门用于满足其协调群集活动的关键角色。在任何情况下都不得将管理器节点部署为生产搜索头。请参阅“管理器节点的其他角色”。

Splunk Enterprise 版本兼容性

不同群集节点类型之间的互操作性会受到兼容性要求的严格限制。简单地说：

- 管理器节点运行的版本必须与对等节点和搜索头的相同或更高。
- 搜索头运行的版本必须与对等节点的相同或更高。
- 所有对等节点运行的版本必须完全相同，至少为维护等级。

管理器节点和对等节点与搜索头之间的兼容性

对等节点和搜索头能运行与管理器节点不同的版本，但受到下面这些限制：

- 管理器节点运行的版本必须与对等节点和搜索头的相同或比它们更高。
- 管理器节点运行的版本最多比对等节点的高三个次要版本。例如，管理器节点 8.0 可以在 7.3、7.2 和 7.1 对等节点上运行，但不能在 7.0 对等节点上运行。
- 所有节点必须运行 7.0 或以上的版本。

对等节点之间的兼容性

所有对等节点必须运行相同版本的 Splunk Enterprise，至少为维护等级。必须同时将所有对等节点更新至最新版本。例如，对于同一索引器群集，不可以部分对等节点的版本为 8.0.2 而另一些对等节点的为 8.0.1。

对等节点和搜索头之间的兼容性

对等节点和搜索节点可以运行不同的版本。搜索头运行的版本必须与对等节点的相同或更高。

索引器群集中的搜索头群集和单个搜索头的兼容性要求相同。有关其他搜索头群集版本要求的信息，请参阅*《分布式搜索》*手册中的“搜索头群集的系统要求和其他部署注意事项”。

计算机要求

群集的每个节点（管理器节点、对等节点和搜索头）必须在自己的单独计算机或虚拟机上运行。除此之外，硬件要求（除了存储）基本上与任何 Splunk Enterprise 实例的相同。请参阅*《容量规划手册》*中的“参考硬件”。

主要差异是对等节点的存储要求，将在下文中说明。

注意：管理器节点的存储需求明显要比“参考硬件”主题中指定的要求低得多，因为管理器节点不会建立外部数据索引。

操作系统要求

Splunk Enterprise 所有支持的操作系统都能使用索引器群集化。对于受支持的操作系统列表，请参阅*《操作手册》*中的“系统要求”。

所有的索引器群集节点（管理器节点、对等节点和搜索头）必须运行相同的操作系统和版本。

如果索引器群集与搜索头群集集成，则搜索头群集实例（包括 Deployer）必须运行与索引器群集节点相同的操作系统及版本。

跨群集系统时钟同步

在所有参与群集活动的运行 Splunk Enterprise 实例的虚拟或物理的计算机上同步系统时钟很重要。具体地说，这意味着管理器节点、对等节点和搜索头。否则，可能出现各种问题，例如在管理器节点和对等节点间的时序问题、搜索失败、或搜索项目的过早失效。

使用的同步方法取决于计算机的具体设置。请查阅运行 Splunk Enterprise 的特定计算机和操作系统的系统文档。对于大多数环境，网络时间协议 (NTP) 是最佳方法。

存储注意事项

当确定群集索引的存储要求时，您需要考虑增加一组对等节点容量以处理多个数据副本。

强烈建议您配置所有对等节点使用相同的磁盘存储量。

群集将使用常用设置管理索引存储，如“配置索引存储”中所述。

确定存储要求

确保具有足够的磁盘空间来容纳对等节点将处理的数据量，这一点很重要。关于 Splunk Enterprise 数据量以及如何评估存储需求的常规讨论，请参阅*《容量规划手册》*中的“评估存储要求”。此主题提供了有关如何评估非群集索引器的存储信息，您需要补充其指南将群集存储的额外数据副本包含在内。

对于群集，除了考虑传入数据量，还必须考虑复制因子和**搜索因子**，以使一组对等节点达到总存储要求。复制因子为 3 时，您将存储三个数据副本。您将需要更多的存储空间来容纳这些副本，但并不需要三倍的存储空间。不可搜索数据的复制副本小于可搜索数据副本，因为它们只包括数据，并不是相关的索引文件。例如，如果复制因子为 3，搜索因子为 2，则与在非群集索引器上存储相同数据相比，将需要大于 2 倍，小于 3 倍的存储容量。

不可搜索副本需要的存储空间具体降低多少，您应进行一些调查。被不可搜索副本排除在外的索引文件大小可能差别很大，具体取决于在*《容量规划手册》*中“评估存储要求”中描述的因子。

重要提示：管理器节点不知道各个对等节点的存储量，因此它在决定哪个对等节点应接收某组特定复制的数据时并不考虑可用的存储空间。它还会随意做出决定，在哪个对等节点上某组复制的数据应是可搜索的（当搜索因子为 2 或更高时）。因此，您必须确保每个对等节点都有足够的存储空间，不仅可以容纳该对等节点上的数据，还可以容纳从其他对等节点流送至此对等节点的任何数据复制副本。您应该在整个群集生命周期内持续监视存储空间使用量。

存储要求示例

作为一个大致估算，传入的 syslog 数据经过压缩和建立索引后，大概会占用其原始大小的 50%：

- 原始数据文件占 15%。
- 关联索引文件占 35%。

实际上，根据*《容量规划手册》*的“评估存储要求”中描述的因子，此估算可能有很大差异。

假设有 100GB 的 syslog 数据流入 Splunk Enterprise。对于非群集索引器，数据大概会在索引器上占用 50GB（100GB 的 50%）的存储空间。但是，对于群集，存储空间的计算必须考虑复制因子和搜索因子，以使所有群集对等节点达到总存储空间要求。（正如前面提到的，无法迅速预测出在任何特定对等节点上需要的确切存储空间量。）

下面列举了两个评估群集存储要求的示例，两个示例均假定传入 syslog 数据为 100GB，结果每组原始数据为 15GB，每组索引文件为 35GB：

- **3 个对等节点，复制因子 = 3；搜索因子 = 2：**这要求所有对等节点的总存储空间达到 115GB（平均 38GB/对等节点），计算如下：
  - 原始数据总计 = (15GB * 3) = 45GB。
  - 总索引文件 = (35GB * 2) = 70GB。
- **5 个对等节点，复制因子 = 5；搜索因子 = 3：**这要求所有对等节点的总存储空间达到 180GB（平均 36GB/对等节点），计算如下：
  - 原始数据总计 = (15GB * 5) = 75GB。
  - 总索引文件 = (35GB * 3) = 105GB。

存储硬件

在 6.0 之前版本的 Splunk Enterprise 中，复制的群集数据桶副本始终驻留在 `colddb` 目录，即使它们是热或温数据桶。自 6.0 版本起，热和温复制的副本驻留在 `db` 目录中，非复制副本也相同。与非群集索引相比，这消除了为群集索引考虑 `colddb` 的更快存储的任何需求。

许可授权信息

如同任意 Splunk Enterprise 部署，您的许可要求由索引器处理的数据量来驱动。请与您的 Splunk 销售代表联系，购买更多的许可量。请参阅*《管理员手册》*中的“许可授权如何运作”，以了解有关 Splunk Enterprise 许可授权的更多信息。

只有几个特定于索引复制的许可证问题：

- 所有群集节点（包括管理器节点、对等节点和搜索头）都需要在 Enterprise **许可证池**中，即使不需要它们对任何数据建立索引。
- 群集节点必须共享相同的许可授权配置。
- 只有传入数据用于计算许可证的数据量，复制的数据不计算在内。
- 使用 Free 许可证时不能使用索引复制。

群集节点使用的端口

对群集节点来说，这些端口必须可用：

- **在管理器节点上：**
  - 管理端口（默认为 8089）必须对所有其他群集节点可用。
- **在每个对等节点上：**
  - 管理端口必须对所有其他群集节点可用。
  - 复制端口必须对所有其他对等节点可用。
  - 接收端口必须对所有发送数据到该节点的转发器可用。
- **在每个搜索头上：**
  - 管理端口必须对所有其他节点可用。
  - http 端口（默认为 8000）必须对所有从搜索头访问数据的浏览器可用。

部署服务器和群集

请勿对群集对等节点使用部署服务器。

不支持通过部署服务器将配置或应用分发到群集对等节点。要跨一组群集对等节点分布配置，请使用“更新通用对等节点配置”主题中概述的**配置软件包**方法。

有关如何将应用分布从部署服务器迁移到配置软件包方法的信息，请参阅“将应用迁移到群集”。

管理器节点的其他角色

正常情况下，您应该将运行在管理器节点的 Splunk Enterprise 实例专用于单一目的。仅可将管理器节点的内置搜索头用于调试。

在受限情况下，您可在管理器节点实例上共存这些轻型功能中的一个或多个：

- **许可证主服务器**
- **监视控制台**
- **搜索头群集 Deployer**

要针对这些额外角色使用管理器节点实例，管理器节点群集必须低于以下限制：

- 30 个索引器
- 100,000 个数据桶
- 10 个索引
- 10 个搜索头

任何情况下，均不得在管理器节点上共存部署服务器。

管理器节点和部署服务器在执行任务时都会占用大量的系统资源。管理器节点需要可靠且可以持续访问资源的权限来持续管理群集，部署服务器在部署更新其部署客户端时可轻松塞满这些资源。

有关管理组件共存的常规讨论，请参阅*《分布式部署手册》*中的“帮助管理部署的组件”。

启用索引器群集管理器节点

阅读此主题前，请先参阅“索引器群集部署概述”。

一个群集有一个且只会有一个**管理器节点**。管理器节点用于协调对等节点的活动。它自身不会存储或复制数据（除其自身内部数据外）。

重要提示：管理器节点无法作为对等节点或搜索节点承担双重功能。作为管理器节点启用的 Splunk Enterprise 实例必须只执行该单一索引器群集角色。另外，管理器节点不能与对等节点共享同一台计算机。然而在受限情况下，管理器节点实例也能处理一些其他轻型功能。请参阅“管理器节点的其他角色”。

在部署群集时必须作为第一个步骤在设置对等节点之前就启用管理器节点。

本主题中的程序说明了如何使用 Splunk Web 启用管理器节点。也可以使用另外两种方法来启用管理器节点：

- 直接编辑管理器节点的 server.conf 文件。有关详细信息，请参阅“使用 server.conf 配置管理器节点”。有些高级设置只能通过编辑此文件进行配置。
- 使用 CLI edit cluster-config 命令。有关详细信息，请参阅“使用 CLI 配置管理器节点”。

重要提示：本主题介绍如何仅为单个站点群集启用管理器节点。如果您打算部署一个多站点群集，请参阅“使用 server.conf 配置多站点索引器群集”。

启用管理器节点

将索引器作为管理器节点启用：

1. 单击 Splunk Web 右上角的**设置**。

2. 在**分布式环境**组中，单击**索引器群集化**。

3. 选择**启用索引器群集化**。

4. 选择**管理器节点**并单击**下一步**。

5. 需要填写几个字段：

- **复制因子**。**复制因子**决定群集可保留多少数据副本。默认值为 3。更多有关复制因子的信息，请参阅“复制因子”。现在请确保选择正确的复制因子。不建议在群集包含大量数据后增加复制因子。
- **搜索因子**。**搜索因子**决定群集保留的可立即搜索数据副本的数量。默认值为 2。更多有关搜索因子的信息，请参阅“搜索因子”。现在请确保选择正确的搜索因子。不建议在群集包含大量数据后增加搜索因子。
- **安全密钥**。这是验证管理器节点与对等节点以及搜索头之间通信的密钥。所有群集节点的密钥必须相同。您在此设置的值必须与随后在对等节点和搜索头上设置的值相同。
- **群集标签**。您可以在此处给群集贴标签。标签对于识别监视控制台中的群集很有用。请参阅*《监视 Splunk Enterprise 手册》*中的“设置群集标签”。

6. 单击**启用管理器节点**。

将显示消息，“您必须重新启动 Splunk 以便管理器节点启动。您可以从服务器控件重新启动 Splunk。”

7. 单击**转到服务器控件**。这将带你前往“设置”页面，可从中执行重新启动。

重要提示：首次启动管理器节点时，它将阻止在对等节点上建立索引，直到您启用并重新启动了个数等于复制因子的全部对等节点为止。当等待对等节点加入群集时不要重启管理器节点。如果重启了主节点，对等节点将需要再次重启。

查看管理器节点仪表板

重新启动后，重新登录到管理器节点，并返回到 Splunk Web 中的“群集化”页面。此时您会看到管理器节点群集化仪表板。有关仪表板的信息，请参阅“查看管理器节点仪表板”。

执行其他配置

请参阅“管理器节点配置概述”，获得关于部署后管理器节点配置的信息。

启用对等节点

阅读此主题前，请先参阅“索引器群集部署概述”。

您通常需要启用多个**对等节点**来部署群集。启用的对等节点数至少等于**复制因子**数，可能更多，以便满足横向扩展的需要。

在启用一组对等节点之前，必须启用并重新启动**管理器节点**，如“启用管理器节点”所述。首次启动管理器节点时，它将阻止在对等节点上建立索引，直到您启用并重新启动个数等于复制因子的对等节点为止。

本主题中的程序说明了如何使用 Splunk Web 启用对等节点。也可以使用另外两种方法来启用对等节点：

- 直接编辑对等节点的 server.conf 文件。有关详细信息，请参阅“使用 server.conf 配置对等节点”。
- 使用 CLI edit cluster-config 命令。有关详细信息，请参阅“使用 CLI 配置对等节点”。

重要提示：本主题介绍如何仅为单个站点群集启用节点。如果您打算部署一个多站点群集，请参阅“使用 server.conf 配置多站点索引器群集”。

启用对等节点

作为对等节点启用索引器：

1. 单击 Splunk Web 右上角的**设置**。

2. 在**分布式环境**组中，单击**索引器群集化**。

3. 选择**启用索引器群集化**。

4. 选择**对等节点**并单击**下一步**。

5. 需要填写几个字段：

- **管理器 URI**。输入管理器节点 URI，包括它的管理端口。例如：https://10.152.31.202:8089。
- **对等节点复制端口**。这是对等节点用来从其他对等节点接收复制的数据的端口。您可以指定将任何可用的、未使用的端口用于此用途。此端口必须不同于管理端口或接收端口。
- **安全密钥**。这是验证管理器节点与对等节点以及搜索头之间通信的密钥。所有群集节点的密钥必须相同。此处设置的值须与之前在管理器节点上设置的值相同。

6. 单击**启用对等节点**。

将显示消息，“您必须重新启动 Splunk 以便对等节点启动。”

7. 单击**转到服务器控件**。这将带你前往“设置”页面，可从中执行重新启动。

8. 为所有群集的对等节点重复此过程。

当您启用了个数等于复制因子的对等节点后，群集便开始建立索引和复制数据，如“启用管理器节点”所述。

查看对等节点仪表板

重新启动后，重新登录到对等节点，并返回到 Splunk Web 中的“群集化”页面。此时您会看到对等节点的群集化仪表板。有关仪表板的信息，请参阅“查看对等节点仪表板”。

配置对等节点

启用对等节点后，在开始为数据建立索引之前需要进一步配置它们。有关详细信息，请阅读以下主题：

- “在索引器群集中配置对等节点索引”
- “将数据导入索引器群集”

可能还需配置对等节点的其他设置。请参阅“对等节点配置概述”。

启用搜索头

阅读此主题前，请先参阅“索引器群集部署概述”。

要搜索群集，需要在索引器群集中至少启用一个**搜索头**。

在启用搜索头前，必须启用并重新启动管理器节点，如“启用管理器节点”中所述。

本主题中的程序说明了如何使用 Splunk Web 来启用搜索头。也可以使用另外两种方法来启用搜索头：

- 直接编辑搜索头的 server.conf 文件。有关详细信息，请参阅“使用 server.conf 配置搜索头”。有些高级设置（包括多群集搜索）只能通过编辑此文件进行配置。
- 使用 CLI edit cluster-config 命令。有关详细信息，请参阅“使用 CLI 配置搜索头”。

重要提示：本主题介绍如何仅为单个站点群集启用个别搜索头：

- 如果您打算部署一个多站点群集，请参阅“使用 server.conf 配置多站点索引器群集”。
- 如果您计划整合作为**搜索头群集**成员的搜索头，请参阅*《分布式搜索》*手册中的“通过索引器群集集成搜索头群集”。

启用搜索头

要在索引器群集中启用 Splunk 实例作为搜索头：

1. 单击 Splunk Web 右上角的**设置**。

2. 在**分布式环境**组中，单击**索引器群集化**。

3. 选择**启用群集化**。

4. 选择**搜索头节点**并单击**下一步**。

5. 需要填写几个字段：

- **管理器 URI**。输入管理器节点 URI，包括它的管理端口。例如：`https://10.152.31.202:8089`。
- **安全密钥**。这是验证管理器节点与对等节点以及搜索头之间通信的密钥。所有群集节点的密钥必须相同。此处设置的值须与之前在管理器节点上设置的值相同。

6. 单击**启用搜索头节点**。

将显示消息，“您必须重新启动 Splunk 以便搜索节点启动。您可以从服务器控件重新启动 Splunk。”

7. 单击**转到服务器控件**。这将带你前往“设置”页面，可从中执行重新启动。

后续步骤

在启用搜索头后，您可以：

- 查看搜索头仪表板
- 允许搜索头搜索其他群集
- 添加搜索头到群集
- 您可在搜索头上进行额外配置

查看搜索头仪表板

重新启动后，重新登录到搜索头，并返回到 Splunk Web 中的“群集化”页面。此时您会看到搜索头的群集化仪表板。请参阅“查看搜索头仪表板”以了解更多信息。

允许搜索头搜索多个群集

从仪表板，您可以添加其他群集以便搜索头搜索。有关详细信息，请参阅“跨多个索引器群集搜索”。

添加搜索头到索引器群集

您可以设置多个搜索头，以适应多个同时进行的搜索。有关如何确定搜索头要求的信息，请参阅《*容量规划*》手册中的“硬件容量规划”。

如果要为单个索引器群集设置多个搜索头，只需对其他实例重复启用过程。如果您想部署搜索头群集使搜索头共享配置和任务，请参阅《*分布式搜索*》手册中“集成搜索头群集和索引器群集”主题中的其他配置说明。

执行其他配置

关于索引器群集中搜索头配置的更多信息，请参阅“搜索头配置概述”。

最佳做法：将管理器节点数据转发到索引器层

我们认为最佳的做法是，将所有管理器节点内部数据转发到索引器（对等节点）层。这有几个优势：

- 它将所有数据累计到一个位置。这简化了管理数据的过程：您只需在一层（索引器层）管理索引和数据。
- 如果管理器节点关闭，它将为管理器节点启用诊断。在故障之前的数据累计在索引器上，其中一个搜索头之后可访问它。

首选方式是将数据直接转发到索引器，无需在管理器节点上单独索引。您可通过将管理器节点配置为转发器来执行此操作。下面是主要步骤：

1. **确保所有必要的索引存在于索引器上。**这是通常的情况，除非您已经在管理器节点上创建了自定义索引器。由于 `_audit` 和 `_internal` 存在于索引器以及管理器节点上，因此您不需要创建这些索引的单独版本以保存相应的管理器节点数据。

2. **将管理器节点配置为转发器。**在管理器节点上创建 `outputs.conf` 文件，为跨对等节点集的负载均衡的转发而配置管理器节点。您也必须关闭管理器节点上的索引，这样管理器节点既不在本地保留数据也不转发数据到对等节点。

以下是一个 outputs.conf 文件示例：

```
# Turn off indexing on the manager node
[indexAndForward]
index = false

[tcpout]
defaultGroup = my_peers_nodes
forwardedindex.filter.disable = true
indexAndForward = false

[tcpout:my_peers_nodes]
server=10.10.10.1:9997,10.10.10.2:9997,10.10.10.3:9997
```

本示例假设每一个对等节点的接收端口都配置为 9997。

有关配置 outputs.conf 的详细信息，请参阅《转发数据》手册中的“使用 outputs.conf 配置转发器”。

准备对等节点进行索引复制

启用对等节点后，可能还需要执行更多配置，以准备对等节点进行索引复制。

如果您仅使用默认的索引集和默认配置，即可开始复制数据。然而，如果需要安装应用或更改对等节点上的配置，则通常必须应用更改集合到所有对等节点。特别是，如果您需要添加索引（包括由应用定义的索引），则必须以确保对等节点使用通用索引集合的方式操作。

有关如何在群集对等节点之间配置索引的信息，请参阅“在索引器群集中配置对等节点索引”。

有关如何在对等节点之间配置应用的信息以及其他对等节点配置问题，请参阅“对等节点配置概述”。

使用索引器群集调整索引

索引器群集的主要用途是启用**索引复制**。但是，即使不需要索引复制，通常也可以在横向扩展部署拓扑中使用群集作为管理多个索引器的一种方式。

例如，假设您要创建一个包含三个索引器和一个搜索头的部署，以使您能够为更大量的数据建立索引，而不是像单个索引器那样只能为少量数据建立索引。执行此操作的常规方式是单独设置每个索引器，添加到搜索头中，然后使用诸如**部署服务器**之类的工具协调索引器配置，这也是在 Splunk Enterprise 5.0 之前的版本唯一可行的方式。

使用群集化，您可以将此部署方案配置为群集，并用三个对等节点替代三个独立索引器。即使您不需要索引复制及其自身的一些重要优点（如数据可用性和灾难容错），使用群集来协调多个索引器实例可能也会很有益处的，有以下几个原因：

- 简化索引器配置的管理和协调（代替使用部署服务器或执行手动更新）。有关详细信息，请参阅“更新通用对等节点配置”。
- 简化分布式搜索的设置和控制。请参阅“启用搜索头”。
- 通过群集化仪表板更好地了解索引器的状态。请参阅“查看管理器节点仪表板”。
- 能够在开发时利用其他群集管理功能。

使用群集调整索引容量的主要缺点如下：

- 必须安装另一个 Splunk Enterprise 实例来充当群集管理器节点。
- 群集不支持异构索引器。所有群集节点必须位于同一版本级别上。此外，群集中的所有对等节点必须使用相同的 indexes.conf 配置。有关更多的详细信息，请参阅下一部分“群集对等节点管理与部署服务器的比较”。
- 不能使用部署服务器在群集对等节点中分布配置或应用。有关更多的详细信息，请参阅下一部分“群集对等节点管理与部署服务器的比较”。

群集对等节点管理与部署服务器的比较

群集能够从一个中央位置（即管理器节点）管理和更新所有索引器（对等节点）的配置，这是一项非常有用的功能。在此方面，群集的功能类似于**部署服务器**。但是与部署服务器不同，群集对等节点管理没有任何服务器类的概念。鉴于这个原因以及群集协调其活动的方式，您不能为不同的索引器组指定不同的应用或 indexes.conf 配置。（群集中的所有对等节点必须使用相同的 indexes.conf 配置和某些其他配置，如“对等节点配置概述”所述。）如果您需要保留一组异构索引器，不能使用群集来实现调整目的。

另一方面，与部署服务器相比，使用配置软件包方法将更新下载到对等节点具有一些优势。具体地说，此方法不仅能分布更新，而且还能在对等节点上验证更新，然后（必要时）以滚动方式重新启动对等节点。有关详细信息，请参阅“更新通用对等节点配置”。

重要提示：不使用部署服务器或第三方分布式配置管理软件，例如 Puppet 或 Chef，来部署对等节点的直接更新。您可以使用这些工具来部署管理器节点的更新，然后管理器节点会部署对等节点的更新。请参阅“使用部署服务器将应用分布到管理器节点”。

配置群集以进行横向扩展部署

要设置群集以进行横向扩展部署，而不进行索引复制，只需将**复制因子**和**搜索因子**同时设置为 1。这会使群集完全充当一组协调的 Splunk Enterprise 实例，而无需数据复制。群集将不会为数据创建任何重复副本，因此您可以将存储大小和处理开销保持在最低限度。

将非群集索引器迁移到群集环境

如果计划将当前的一组索引器迁移至索引器群集，请仔细阅读主题“群集和非群集索引器部署之间的关键差异”。该主题介绍了您在启动迁移过程之间必须了解的问题。

可以随时将非群集索引器添加到群集（作为一个对等节点）。要这样做，只需按“启用对等节点”中所述，作为对等节点启用索引器。

一旦将索引器启用为对等节点，它就会与其他任何对等节点一样参与到群集中。任何传入对等节点的新数据都会根据群集的复制因子得以复制，此对等节点也是一个从其他对等节点接收复制的数据的候选节点。系统不会自动复制索引器上已存在的数据，但这些数据确实会根据如下所述参与搜索。

重要提示：在将一个索引器从非群集迁移到群集之前，请确认您的需求。此过程只能单向进行。不支持索引器从群集转换成非群集的过程。

管理旧数据

“旧数据”一词是指在索引器在转换为群集对等节点之前已经存储的数据。

群集如何处理旧的索引数据

向群集添加现有索引器时，群集不会复制索引器中已存在的任何数据桶。

在添加到群集之前索引器中的已有数据桶称为“独立”数据桶。搜索仍在这些数据桶中进行，并会与群集的复制数据桶的搜索结果相结合。

有方法可以迁移我的旧数据吗？

由于将独立数据桶转换为复制数据桶的处理成本很高（因为需要为这些数据桶创建多个可搜索和不可搜索副本，以与群集的复制因子和搜索因子相一致），所以这样做是不明智之举，尤其在索引器具有大量独立数据桶的情况下。没有受支持的过程来实现这种转换。但是，如果十分需要执行这一转换，请联系 Splunk 专业服务人员，探讨此操作的利弊和要求。

将应用迁移到群集

在您当前的非群集环境中，可能使用**部署服务器**将应用分发给一组索引器。将索引器转换为群集对等节点后，将无法再次进行如此操作。有关详细信息，请参阅“群集和非群集索引器部署之间的关键差异”。

要将应用分发给对等节点，必须按照“更新通用对等节点配置和应用”所述改用**配置软件包**方法。将应用存放在管理器节点上的特殊位置，然后管理器节点在首次启用索引器作为对等节点时将应用推送到各个索引器。如果以后需要添加或更改应用，则在管理器节点上进行更改和添加，然后告诉管理器节点将已更新的配置软件包推送到整组对等节点。

有关配置软件包方法与部署服务器方法比较的信息，请参阅“群集对等节点管理与部署服务器的比较”。

重要提示：*必须*使用配置软件包方法将应用分发给对等节点；不支持使用部署服务器执行此任务。

如何迁移应用

建议您在启用群集时迁移应用。下面介绍的过程具体说明了操作步骤。

重要提示：在您尝试执行此过程之前，您必须熟悉当前章节中前面介绍的各个主题。请从“索引器群集部署概述”开始，一直阅读直到返回本主题。

此过程假定您从分布式搜索环境开始，并将一组索引器配置为部署服务器的部署客户端。使用部署服务器将应用推送到索引器。您将转换为群集对等节点的就是这些索引器。当变成对等节点后，您就不能再使用部署服务器将应用推送给它们；而必须改用配置软件包方法。

要在启用群集时迁移应用，应遵循以下步骤：

1. 按照“启用管理器节点”所述启用管理器节点。

2. 在部署服务器上，找到推送到您计划迁移的索引器的一组应用。这些应用应该在部署服务器的 $SPLUNK_HOME/etc/deployment-apps 目录下。

3. 将这些应用从部署服务器复制到群集管理器节点的 $SPLUNK_HOME/etc/master-apps 目录中。有关 master-apps 目录的信息，请参阅“配置软件包的结构”。

4. 检查 master-apps 中的每个应用是否有 indexes.conf 文件。在这些文件中，找到定义新索引的段落。在每个段落中，添加以下属性/值对：

```
repFactor=auto
```

这样便为索引启用了复制。有关更多信息，请参阅“indexes.conf repFactor 属性”。

注意：如果您是应用的创建者和维护者，还可以在 $SPLUNK_HOME/etc/master-apps/<appname>/default/indexes.conf 中进行此更改。

5. 将每个索引器转换为群集对等节点，一次转换一个：

   a. 重新配置索引器，使其不再是部署客户端。

   b. 删除索引器的 $SPLUNK_HOME/etc/apps 目录中的部署应用。保留默认应用，如搜索。

   c. 启用索引器作为群集对等节点，如“启用对等节点”所述。

   d. 重新启动对等节点，完成启用过程。

   e. 验证管理器节点已将需要的一组应用推送到对等节点的 $SPLUNK_HOME/etc/slave-apps 目录。

6. 启用所有对等节点后，转到管理器节点仪表板，并验证正在复制需要的一组索引。此仪表板在“查看管理器节点仪表板”中有介绍。

有关配置群集对等节点的更多信息，请参阅以下主题：

- “在索引器群集中配置对等节点索引”
- “对等节点配置概述”
- “更新通用对等节点配置和应用”

索引器群集升级

由于升级性质的不同，升级过程也有所差异。本主题介绍如下基于版本的情况：

- 从 7.1 或更高版本升级
- 从 6.x 升级到 7.1
- 升级到新维护版本（例如，从 8.1.1 到 8.1.2）

此外，本主题介绍：

- 如何升级与搜索头群集集成的索引器群集。
- 如何执行多站点索引器群集的逐个升级。

从单个站点迁移到多站点？

要将单站点索引器群集转换成多站点，请先升级，然后阅读“将索引器群集从单个站点迁移到多站点”。

升级与搜索头群集集成的索引器群集？

如果您正在从 6.x 或更高版本升级，您可以遵循分层升级的步骤单独升级每个群集。请参阅“单独升级每一层”。

否则，您必须同时升级两个群集：

1. 关于适合您部署的索引器群集升级类型，请遵循本主题中的步骤。
2. 在索引器群集升级（需要停止搜索头）的步骤中，停止所有搜索头群集成员。
3. 在索引器群集升级（需要升级搜索头）的步骤中，遵循*《分布式搜索》*中“执行非滚动升级”所介绍的升级过程，执行搜索头群集升级步骤的剩余部分。

升级没有自定义安全密钥的索引器群集？

安全密钥（也称为 pass4SymmKey 设置）会验证管理器节点与对等节点以及搜索头之间通信。

从 6.6 开始，需要非默认的安全密钥。如果群集安全密钥从未明确设置为自定义值，管理器节点上会显示一条警告消息：

```
pass4SymmKey setting in the clustering or general stanza of server.conf is set to empty or the default value. You must change it to a different value.
```

要补救此情况，群集关闭后，您必须在所有群集节点（管理器节点、对等节点、搜索头）上设置安全密钥。所有群集节点的密钥必须相同。

您可以用 server.conf 中的 pass4SymmKey 属性设置安全密钥。请参阅“配置安全密钥”。

要在群集升级期间设置密钥，您必须按照“同时升级所有层”中的流程同时升级所有群集层。所有节点关闭后，设置安全密钥，这样节点启动时会拥有相同的安全密钥。

滚动升级 7.1 或更高版本索引器群集

升级 7.1 或更高版本索引器群集时，首选方法是执行滚动升级。滚动升级方法使您可以将对等节点升级到新版本，同时尽量避免正在进行的搜索中断。请参阅“对 Splunk Enterprise 索引器群集进行滚动升级”。

在单一操作中升级所有对等节点

使用此方法，您可以在升级操作中同时关闭所有对等节点。您可以同时升级群集的所有层（主节点、搜索头、对等节点），也可以分别升级每一层。

升级 6.x 群集时有必要使用此方法。但是，这也可以用于升级任何版本的群集。它的缺点是整个对等节点层都需要停机，这会影响在此期间的索引和搜索，

注意：如果您有多站点群集，大部分情况下，您可以改为一次升级一个群集。请参阅“多站点索引器群集按站点逐一升级”。

如果搜索头层由搜索头群集组成，则分别升级各层的方法是非常有用的，因为它不需要同时升级搜索头群集与索引器群集对等节点。

警告：强烈建议您即使单独升级每层时，仍然要快速完成整个升级过程，以避免运行不同版本的节点类型之间任何不兼容的可能性。

要同时升级所有群集层，请参阅“同时升级所有层”。

要单独升级每一层，请参阅“单独升级每一层”。

同时升级所有层

请执行以下步骤：

1. 停止管理器节点。
2. 停止所有对等节点和搜索头。
   当停止对等节点时，使用 splunk stop 命令而不是 splunk offline。
3. 如果群集没有使用非默认（自定义）安全密钥，您必须立即设置一个。从 6.6 开始，索引器群集需要非默认的安全密钥。群集中所有节点的密钥必须相同。请参阅“升级没有自定义安全密钥的索引器群集？”。在每个节点（管理器节点、对等节点和搜索头）上，使用“配置安全密钥”中的程序设置密钥。
4. 升级管理器节点，遵照任何 Splunk Enterprise 升级的正常程序，如同*《安装手册》*中的“如何升级 Splunk Enterprise”所述。**先不要升级节点。**
5. 如果管理器节点还没有运行，则启动管理器节点，接受所有提示。
6. 在管理器节点上运行 splunk enable maintenance-mode。要确认管理器节点处于维护模式，运行 splunk show maintenance-mode。本步骤可以防止不必要的数据桶修复。请参阅“使用维护模式”。
7. 升级搜索头，随后是对等节点。使用任何 Splunk Enterprise 升级的正常程序，如*《安装手册》*中的“如何升级 Splunk Enterprise”所述。
8. 如果对等节点和搜索头还没有运行，启动它们。
9. 在管理器节点上运行 splunk disable maintenance-mode。要确认管理器节点不处于维护模式，运行 splunk show maintenance-mode。

您可以查看管理器节点仪表板，以确认所有群集节点已经启用并正在运行。

单独升级每一层

当单独升级层时：

- 您必须按照规定的顺序升级层。
- 在每一层中，您必须将升级所有节点作为单个操作进行。

直到所有层均完成升级后新版本中的功能才可用。

警告：强烈建议您即使单独升级每层时，仍然要快速完成整个升级过程，以避免运行不同版本的节点类型之间任何不兼容的可能性。

当升级层以分离操作进行时，必须遵循此升级顺序：

1. 升级管理器节点。
2. 升级搜索头层。
3. 升级对等节点层。

1. 升级管理器节点

1. 停止管理器节点。
2. 升级管理器节点，遵照任何 Splunk Enterprise 升级的正常程序，如同*《安装手册》*中的“如何升级 Splunk Enterprise”所述。
3. 如果管理器节点还没有运行，则启动管理器节点，接受所有提示。

您可以查看管理器节点仪表板，以确认所有群集节点已经启用并正在运行。

2. 升级搜索头层

用于升级搜索头层的方法取决于层是否由搜索头群集组成：

- 如果搜索头层由搜索头群集组成，请遵循“升级搜索头群集”中的步骤。如果需要，您可以执行搜索头群集的滚动升级，如该主题中所述。

- 如果搜索头层由独立的搜索头组成，请遵循以下步骤：

1. 停止所有搜索头。
2. 升级搜索头，遵照任何 Splunk Enterprise 升级的正常程序，如同*《安装手册》*中的“如何升级 Splunk Enterprise”所述。
3. 如果搜索头还没有运行，启动它们。

您可以查看管理器节点仪表板，以确认所有群集节点已经启用并正在运行。

3. 升级对等节点层

1. 在管理器节点上运行 `splunk enable maintenance-mode`。
   要确认管理器节点处于维护模式，在管理器节点上运行 `splunk show maintenance-mode`。
   本步骤可以防止不必要的数据桶修复。请参阅“使用维护模式”。
2. 停止所有对等节点。
   当停止对等节点时，使用 `splunk stop` 命令而不是 `splunk offline`。
3. 升级对等节点，遵照任何 Splunk Enterprise 升级的正常程序，如同*《安装手册》*中的“如何升级 Splunk Enterprise”所述。
4. 如果对等节点还没有运行，启动它们。
5. 在管理器节点上运行 `splunk disable maintenance-mode`。
   要确认管理器节点不处于维护模式，在管理器节点上运行 `splunk show maintenance-mode`。

您可以查看管理器节点仪表板，以确认所有群集节点已经启用并正在运行。

多站点索引器群集按站点逐一升级

警告：您无法使用此方法将带有指标数据的群集从 7.3.x 或更早版本升级为 8.0.x 或更高版本，因为 8.0 中对指标数据的存储方式进行了更改。或者，您必须使用“对 Splunk Enterprise 索引器群集进行滚动升级”中介绍的滚动升级方法或者将关闭和升级所有对等节点作为单个操作进行，如“在单个操作中升级所有对等节点”中介绍的方法所述。正如您在研究这些方法时注意到的那样，在对等节点升级期间，它们都将集群置于维护模式。此步骤至关重要，因为它确保在 7.3.x 对等节点和 8.0.x 对等节点之间不会发生指标数据复制。

如果您有一个多站点群集，您可以一次升级一个站点，只要升级的版本之间不超过一个序列 *n.n* 版本（例如，可以从 6.5 到 6.6、6.6 到 7.0、但不能从 6.5 到 7.0）。因为每个站点都有一套完整的主要副本，这方法使得在升级期间搜索的运行不会中断。

如果升级的版本之间超过一个序列 *n.n* 版本（例如，从 6.4 到 6.6 或者从 6.5 到 7.0），则不能执行逐站升级。要跨多个序列 *n.n* 版本升级，您必须使用“对 Splunk Enterprise 索引器群集进行滚动升级”中介绍的滚动升级方法或者将关闭和升级所有对等节点作为单个操作进行，如“在单个操作中升级所有对等节点”中介绍的方法所述。

或者，要跨多个序列 *n.n* 版本升级，您可以通过不超过一个序列 *n.n* 版本的过渡版本升级。例如，如果您从 6.4 升级到 6.6，您可以首先使用逐站升级的方法升级到 6.5 过渡版本，然后再升级到 6.6。

直到所有节点均完成升级后新版本中的功能才可用。

对于双站点群集，升级过程有三个不同的阶段：

1. 升级管理器节点。

2. 升级 site1 对等节点和搜索头。

3. 升级 site2 对等节点和搜索头。

具体步骤如下：

1. 停止管理器节点。

2. 升级管理器节点，遵照任何 Splunk Enterprise 升级的正常程序，如同*《安装手册》*中的“如何升级 Splunk Enterprise”所述。

3. 如果管理器节点还没有运行，则启动管理器节点，接受所有提示。

4. 在管理器节点上运行 `splunk enable maintenance-mode`。要确认管理器节点处于维护模式，运行 `splunk show maintenance-mode`。本步骤可以防止不必要的数据桶修复。请参阅“使用维护模式”。

5. 运行 `splunk stop` 命令停止 site1 上的所有对等节点和搜索头。

6. 升级 site1 对等节点和搜索头。

7. 如果 site1 对等节点和搜索头还没有运行，则启动它们。

8. 在管理器节点上运行 `splunk disable maintenance-mode`。要确认管理器节点不处于维护模式，运行 `splunk show maintenance-mode`。

9. 等待一段时间，直到管理器节点仪表板上显示搜索因子和复制因子均满足条件。

10. 在管理器节点上运行 `splunk enable maintenance-mode`。要确认管理器节点处于维护模式，运行 `splunk show maintenance-mode`。

11. 运行 `splunk stop` 命令停止 site2 上的所有对等节点和搜索头。

12. 升级 site2 对等节点和搜索头。

13. 如果 site2 对等节点和搜索头还没有运行，则启动它们。

14. 在管理器节点上运行 `splunk disable maintenance-mode`。要确认管理器节点不处于维护模式，运行 `splunk show maintenance-mode`。

您可以查看管理器节点仪表板，以确认所有群集节点已经启用并正在运行。

升级到维护版本

要升级群集到维护版本（例如，从 8.1.1 至 8.1.2），则无需立即关闭整个群集。与此相反，您可以执行滚动、在线升级，其中可一次升级一个节点。

警告：即便有滚动升级，您还是应该迅速地升级所有节点，有以下几个原因：

- 群集的正常运行取决于运行在 Splunk Enterprise 同一版本上的所有对等节点，正如在“索引器群集的系统要求和其他部署注意事项”中所述。
- 其他版本兼容性要求也必须满足，如在“索引器群集的系统要求和其他部署注意事项”中所述。
- 如果您升级管理器节点，但不升级对等节点，群集可能会生成错误和警告。如果时间较短，尚可允许，但仍应尽快完成所有节点的升级。

要升级群集节点，除了如下所述的几个例外，请按 Splunk Enterprise 升级的正常过程操作。有关升级 Splunk Enterprise 实例的一般信息，请参阅“如何升级 Splunk Enterprise”。

要执行滚动维护升级，请遵循以下步骤：

1. 升级管理器节点

首先升级管理器节点。

有关管理器节点发生故障时会出现什么情况以及管理器节点恢复后会出现什么情况的信息，请参阅“管理器节点故障时的情况”。

2. 升级搜索头

升级搜索头时对群集的唯一影响是升级期间会中断搜索。

3. 使管理器节点进入维护模式

在管理器节点上运行 `splunk enable maintenance-mode`。要确认管理器节点处于维护模式，运行 `splunk show maintenance-mode`。本步骤可以防止不必要的数据桶修复。请参阅“使用维护模式”。

4. 升级对等节点

升级对等节点时，请注意以下几点：

- 对等节点升级会中断正在执行的搜索。
- 为了将停机时间降到最少，并限制建立索引和搜索的中断情况，升级对等节点时请每次升级一个。
- 要在升级前停止对等节点，使用 `splunk offline` 命令，如“使对等节点脱机”所述。
- 在升级管理器节点和完成升级对等节点之间的过渡期间，群集可能生成各种警告和错误。
- 对于多站点群集，对等节点升级的站点顺序无关紧要。

5. 使管理器节点退出维护模式

在管理器节点上运行 `splunk disable maintenance-mode`。要确认管理器节点不处于维护模式，运行 `splunk show maintenance-mode`。

执行索引器群集的滚动升级

Splunk Enterprise 7.1.0 及更高版本支持索引器群集的滚动升级。滚动升级使您可以对索引器对等节点进行分段升级，尽量减少中断不间断搜索。在将对等节点升级到新版本的 Splunk Enterprise 时，您可以使用滚动升级尽量减少搜索中断。

要求和注意事项

开始滚动升级之前查看以下要求和注意事项：

- 滚动升级只适用于从 Splunk Enterprise 7.1.x 升级到更高版本。
- 管理器节点和所有对等节点必须运行 7.1.0 或更高版本。关于升级说明，请参阅“升级索引器群集”。
- 所有搜索头和搜索头群集必须运行 7.1.0 或更高版本。
- 升级过程中请勿尝试任何群集化维护操作，如滚动升级、软件包推送或节点增设。

导致无法关闭或重新启动节点的硬件或网络故障可能需要手动干预。

滚动升级如何工作

启动滚动升级之后，您可以选择对等节点然后使其脱机。脱机过程中，管理器节点会将数据桶主要副本重新分配到其他对等节点，以保持可搜索状态，对等节点会在可配置超时值内完成任何正在进行的搜索。请参阅“快速版的脱机过程”。

管理器节点关闭对等节点后，在对等节点重新加入群集时，执行软件升级并使对等节点恢复联机。您可以对每个对等节点重复此过程，直到完成滚动升级。

滚动升级的行为方式如下所示：

- 根据默认搜索因子 SF=2，一次只升级一个对等节点。如果 SF 为 3 或更大，您一次可升级 SF-1 个对等节点。例如，如果 SF=3，您一次可升级 2 个对等节点。对单站点群集和多站点群集来说，您可以同时升级的对等节点数量是一样的，因为多站点集群的指导是一次升级一个站点。所以对于多站点群集，如果 SF=3，您可以在相同的站点同时升级 2 个对等节点。
- 对等节点会等待正在进行的搜索完成。最长等待时间由 `server.conf` 中的 `decommission_search_jobs_wait_secs` 属性确定。在大多数情况下，默认 180 秒对大多数搜索来说时间已足够。
- 滚动升级适用于历史搜索和实时搜索。

搜索时间超出（默认为 180 秒）的正在进行的搜索可能会产生不完整的结果和响应的错误消息。如果您有必须完成的计划搜索，请增加 `decommission_search_jobs_wait_secs` 值或在搜索时间范围内不要进行滚动升级。

在您进行滚动升级之前，确保 [search] 段落中 limits.conf 的 search_retry 属性已设为 false（默认）。将此属性设为 true 可能导致搜索时间超过 decommission_search_jobs_wait_secs 值，产生重复或部分结果，但没有错误信息。

禁用延迟的计划搜索

默认情况下，滚动升级期间，根据 savedsearches.conf 中的默认 defer_scheduled_searchable_idxc 属性，连续的计划搜索会延迟，直到升级完成。无论如何设置，实时计划搜索会延迟。

您可禁用此默认行为，这样连续的计划搜索就不会延迟，如下所示：

1. 在搜索头上，编辑 $SPLUNK_HOME/etc/system/local/savedsearches.conf。
2. 将 defer_scheduled_searchable_idxc 设置为 false。

   ```
   [default]
   defer_scheduled_searchable_idxc = false
   ```
3. 重新启动 Splunk。

禁用 defer_scheduled_searchable_idxc 之后，已保存的计划搜索可能会返回部分结果。

有关 defer_scheduled_searchable_idxc 的更多信息，请参阅*管理员手册*中的 savedsearches.conf。

关于实时计划搜索和连续计划搜索的信息，请参阅“实时计划和连续计划”。

执行滚动升级

要在尽量减少搜索中断的情况下升级索引器群集，请执行以下步骤：

1. 运行初级运行状况检查

在管理器节点上，使用 verbose 选项运行 splunk show cluster-status 命令，以确认群集处于可搜索状态：

```
splunk show cluster-status --verbose
```

此命令显示关于群集状态的信息。启动滚动升级之前，查看命令输出以确认是否满足搜索因子，所有数据是否均可搜索。

群集必须有两份每个处于可搜索状态的数据桶的可搜索副本才能进行滚动升级。

此示例为 splunk show cluster-status --verbose 命令输出的内容：

```
splunk@manager1:~/bin$ ./splunk show cluster-status --verbose

Pre-flight check successful .................. YES
 ├──────── Replication factor met ............... YES
 ├──────── Search factor met ..................... YES
 ├──────── All data is searchable ............... YES
 ├──────── All peers are up ..................... YES
 ├──────── CM version is compatible ............. YES
 ├──────── No fixup tasks in progress ........... YES
 └──────── Splunk version peer count { 7.1.0: 3 }

 Indexing Ready YES

 idx1          0026D1C6-4DDB-429E-8EC6-772C5B4F1DB5        default
         Searchable YES
         Status  Up
         Bucket Count=14
         Splunk Version=7.1.0

 idx3          31E6BE71-20E1-4F1C-8693-BEF482375A3F        default
         Searchable YES
         Status  Up
         Bucket Count=14
         Splunk Version=7.1.0

 idx2          81E52D67-6AC6-4C5B-A528-4CD5FEF08009        default
         Searchable YES
         Status  Up
```

```
Bucket Count=14
Splunk Version=7.1.0
```

输出表明运行状况检查成功，即表示群集处于可搜索状态，可进行滚动升级。

有关运行状态检查标准的信息，请参阅“运行状况检查输出详细信息”。

运行状况检查不会涵盖所有潜在的群集运行状况问题。检查只适用于所列出的条件。

或者将 GET 请求发送至以下端点以监视群集运行状况：

```
cluster/manager/health
```

如果端点输出显示为 `pre_flight_check: 1`，则运行状况检查成功。

有关端点的详细信息，请参阅*《REST API 参考手册》*中的 cluster/manager/health。

2. 升级管理器节点

1. 停止管理器节点。
2. 升级管理器节点，遵循 Splunk Enterprise 升级流程。请参见*安装手册*中的“如何升级 Splunk Enterprise”。
3. 如果群集管理器节点还没有运行，则启动管理器节点，并接受所有提示。

您可以使用群集管理器节点仪表板，以确认所有群集节点已经启用并正在运行。请参阅“查看管理器节点仪表板”。

3. 升级搜索头层

如果搜索头层由独立的搜索头组成，请遵循以下步骤：

1. 停止所有搜索头。
2. 升级搜索头，遵照任何 Splunk Enterprise 升级的正常程序，如同*《安装手册》*中的“如何升级 Splunk Enterprise”所述。
3. 如果搜索头还没有运行，启动它们。

如果搜索头层由搜索头群集组成，请遵循“升级搜索头群集”中的步骤。

4. 初始化滚动升级

对管理器节点运行以下 CLI 命令：

```
splunk upgrade-init cluster-peers
```

或者将 POST 请求发送至以下端点：

```
cluster/manager/control/control/rolling_upgrade_init
```

此操作将启动滚动升级并将群集设置为维护模式。

有关端点的详细信息，请参阅*《REST API 参考手册》*中的 cluster/manager/control/control/rolling_upgrade_init。

5. 使对等节点脱机

同时使多个对等节点脱机会影响搜索。

对对等节点运行以下 CLI 命令：

```
splunk offline
```

或者将 POST 请求发送至以下端点。

```
cluster/peer/control/control/decommission
```

管理器节点会重新分配数据桶主要副本，完成任何正在进行的搜索，然后关闭对等节点。

有关端点的详细信息，请参阅*《REST API 参考手册》*中的 cluster/peer/control/control/decommission。

（可选）监视对等节点状态

要监视脱机过程的状态，将 GET 请求发送至以下端点：

```
cluster/manager/peers/<peer-GUID>
```

如果响应显示为 "ReassigningPrimaries"，则表明对等节点尚未关闭。

有关端点的详细信息，请参阅*《REST API 参考手册》*中的 cluster/manager/peers/{name}。

6. 升级对等节点

升级对等节点，遵循 Splunk Enterprise 标准升级流程。请参见*安装手册*中的“如何升级 Splunk Enterprise”。

7. 使对等节点恢复联机

对对等节点运行以下命令。

```
splunk start
```

对等节点启动，然后自动重新加入群集。

8. 验证版本升级

使用以下端点验证版本升级：

```
cluster/manager/peers/<peer-GUID>
```

有关端点的详细信息，请参阅*《REST API 参考手册》*中的 cluster/manager/peers/{name}。

9. 重复步骤 5-8

重复步骤 5-8，直到所有对等节点升级完成。

10. 完成滚动升级

对管理器节点运行以下 CLI 命令：

```
splunk upgrade-finalize cluster-peers
```

或者将 POST 请求发送至以下端点：

```
cluster/manager/control/control/rolling_upgrade_finalize
```

此操作将完成升级程序，并使群集退出维护模式。

有关端点的详细信息，请参阅*《REST API 参考手册》*中的 cluster/manager/control/control/rolling_upgrade_finalize。

冲突操作

您不能同时运行某些操作：

- 数据重新平衡
- 删除多余的数据桶
- 滚动重新启动
- 滚动升级

如果您触发其中一项操作，同时另一项操作也正在运行，那么 splunkd.log、CLI 和 Splunk Web 都会遇到问题，错误讯息指示当前存在冲突操作。

将数据导入索引器群集

将数据导入索引器群集的方式

群集对等节点可以直接从任何与非群集索引器相同的来源获取其数据。但是，如果数据保真度对于您非常重要，则在将数据转发到对等节点之前，应先使用**转发器**以首次使用该数据，而不要将数据直接插入节点。

转发器用于数据导入的优势

使用转发器将数据发送到群集有几个重要原因：

- **确保为所有传入数据建立索引。**通过激活转发器的可选**索引器确认**功能，您可以确保所有传入数据都建立索引并存储在群集中。如果使用索引器确认功能，当源对等节点接收到来自转发器的数据块，它会在对数据建立索引并将数据成功复制到目标对等节点之后，向转发器发送一个确认。如果转发器收到源对等节点发送的确认，则会重新发送数据。转发器会继续重新发送数据，直到其获得确认。索引器确认是确保端到端的数据保真度的唯一方式。请参阅“索引器确认如何工作”。
- **处理可能的节点故障。**使用**负载均衡**的转发器时，如果组中的一个节点故障，则转发器会继续将其数据发送到组中的剩余对等节点。反之，如果您使用直接导入到对等节点，则当其接收对等节点故障时，数据来源就不能继续发送数据。请参阅“负载均衡如何工作”。
- **要简化数据源与对等节点的连接过程。**通过在转发器上启用**索引器发现**，转发器会自动在所有可用的对等节点（包括后来添加到群集中的所有对等节点）间进行负载均衡。请参阅“索引器发现方法的优势”。

直接在对等节点上配置输入

如果您决定不使用转发器来处理数据导入，可以用常规方式（例如，编辑对等节点上的 `inputs.conf`）在每个对等节点上设置输入。有关配置输入的信息，请参阅*《数据导入手册》*中的“配置输入”。

使用转发器将数据导入索引器群集

使用转发器与索引器群集的主要原因是：

- **确保为所有传入数据建立索引。**通过激活转发器的可选**索引器确认**功能，您可以确保所有传入数据都建立索引并存储在群集中。请参阅“索引器确认如何工作”。
- **处理可能的节点故障。**使用**负载均衡**的转发器时，如果组中的一个节点故障，则转发器会继续将其数据发送到组中的剩余对等节点。请参阅“负载均衡如何工作”。
- **要简化数据源与对等节点的连接过程。**通过在转发器上启用索引器发现，转发器会自动在所有可用的对等节点（包括后来添加到群集中的所有对等节点）间进行负载均衡。请参阅“索引器发现方法的优势”。

要使用转发器将数据导入群集中，必须执行两种类型的配置：

- 连接转发器与对等节点。
- 配置转发器的数据导入。

在继续之前，您必须熟悉转发器以及如何使用转发器将数据导入 Splunk Enterprise。有关转发器的介绍，请参阅《*转发数据*》手册中的“关于转发和接收”。该手册中的后续主题介绍了部署和配置转发器的所有方面。

连接转发器与对等节点

有两种方法来连接转发器与对等节点：

- **使用索引器发现功能。**通过使用**索引器发现**，每个转发器会查询管理器节点，以获得群集中所有对等节点的列表。然后它使用负载均衡将数据转发给该组对等节点。如果是在多站点群集中，转发器可以选择查询管理器节点，以获得单个站点上所有对等节点的列表。请参阅“使用索引器发现来连接转发器与对等节点”。
- **直接连接转发器与对等节点。**这是建立转发器/索引器连接的传统方法。您可以直接在转发器上指定对等节点作为**接收器**。请参阅“直接连接转发器与对等节点”。

索引器发现方法的优势

索引器发现比传统方法更有优势：

- 当新的对等节点加入群集时，您不需要重新配置和重新启动转发器来连接到新的对等节点。转发器会自动从管理器节点获

得对等节点的更新列表。它使用负载均衡向列表中的所有对等节点进行转发。

- 您可以添加新的转发器，而无需确定当前的一组群集对等节点。您只需要在新的转发器上配置索引器发现。
- 当您在一组对等节点间转发数据时，可以使用**加权负载均衡**。通过使用索引器发现，管理器节点可以跟踪每个对等节点的磁盘使用总量，并将该信息通知给转发器。然后，转发器根据磁盘容量调整它们发送给每个对等节点的数据量。

配置每个转发器的数据导入

在使用您喜欢的方法指定转发器与接收对等节点之间的连接之后，必须指定每个转发器的数据导入，以使转发器拥有要发送到群集的数据。通常，通过编辑转发器的 inputs.conf 文件来执行此操作。

有关配置数据导入的详细信息，请阅读《*数据导入*》手册（从“Splunk 可为哪些内容创建索引”开始）。该手册中标题为“使用转发器”的主题介绍了如何在转发器上指定数据导入。

索引器确认如何工作

为确保端到端的数据保真度，您必须在要向群集发送数据的每个转发器上明确地显示启用索引器确认。

简单地说，索引器确认的工作方式类似如下所示：转发器以大约 64kB 块的方式不断向接收对等节点发送数据。转发器会在内存中为每个块保留一份副本，直到它收到对等节点的确认为止。等待期间，转发器还会继续发送更多的数据块。

如果一切顺利，接收对等节点会：

1. 接收数据块，对其进行分析并建立数据索引，然后将数据（原始数据和索引数据）写入文件系统。
2. 将原始数据的副本流送至其每个目标对等节点，以满足复制因子。
3. 接收来自每个目标对等节点的通知，不管写入是否成功。
4. 将确认发回到转发器。

确认可确保转发器已成功将数据写入群集中。收到确认后，转发器便会从内存中释放数据块。

如果转发器未收到确认，这意味着故障持续存在。接收对等节点出现故障，或者对等节点无法联系其目标对等节点集合。然后，转发器会重新发送数据块。如果转发器使用负载均衡，则它会将块发送到负载均衡组中的另一个接收节点。如果转发器未设置负载均衡，则它会尝试和以前一样将数据发送到相同节点。

有关索引器确认如何工作的更多信息，请参阅*《转发数据》*手册中的“防止传输中的数据丢失”。

负载均衡如何工作

使用负载均衡时，转发器可在多个接收对等节点之间分布传入数据。每个节点都会获得其中一部分数据，所有接收节点将获得全部数据。

Splunk 转发器会执行“自动负载均衡”。转发器根据指定时间间隔将数据路由到不同节点。例如，假设有一个包含三个对等节点的负载均衡组：A、B 和 C。转发器按 autoLBFrequency 属性（位于 outputs.conf）指定的时间间隔（默认为 30 秒），将数据流切换到组中的另一个节点，数据是随机选择的。因此，每 30 秒，转发器可能从节点 B 切换到节点 A，再切换到节点 C，依此类推。如果某一节点故障，转发器会立即切换到另一个节点。

注意：为了略微详述这一点，每个转发器的输入都具有一个数据流。如果这么做是安全的话，转发器会按指定时间间隔将此数据流切换到新选择的节点。如果转发器无法将此数据流安全切换到新节点，它会使与前一个节点的连接保持打开状态，并继续发送此数据流到该节点，直到该数据流被安全发送出去。

负载均衡与索引器确认结合使用，在群集部署中非常重要，因为它可帮助确保在发生节点故障时不会丢失任何数据。如果转发器没有从它将数据发送到的节点收到索引器确认，它会将数据重新发送到负载均衡组中的下一个可用节点。

使用索引器发现功能的转发器总是使用负载均衡将数据发送给一组对等节点。您可以启用加权负载均衡，这意味着转发器会基于每个对等节点的磁盘容量分发数据。例如，使用 400GB 磁盘的对等节点接收到的数据会是使用 200GB 磁盘的对等节点接收数据的两倍。请参阅“使用加权负载均衡”。

有关更多信息：

- 关于使用索引器发现的负载均衡，请参阅“使用索引器发现来连接转发器与对等节点”。
- 关于不使用索引器发现的负载均衡，请参阅*《转发数据》*手册中的“设置负载均衡”。
- 关于负载均衡如何与索引器确认结合使用，请参阅*《转发数据》*手册中的“防止传输中的数据丢失”。

使用索引器发现来连接转发器与对等节点

索引器发现简化了索引器群集中连接转发器与对等节点的过程。它简化了索引器群集的设置和维护。请参阅“索引器发现方法的优势”。索引器发现只能用于转发到索引器群集。

每个转发器会查询管理器节点，以获得群集中所有对等节点的列表。然后它使用负载均衡将数据转发给该组对等节点。如果是在多站点群集中，转发器可以查询管理器节点，以获得单个站点上所有对等节点的列表。

索引器发现如何工作

简单地说，处理过程的工作方式如下：

1. 对等节点在其接收端口上提供信息給管理器节点。

2. 转发器会按固定的时间间隔轮询管理器节点，以获取可用对等节点的列表。您可以调整该时间间隔。请参阅“调整轮询频率”。

3. 管理器节点将对等节点的 URI 和接收端口传输给转发器。

4. 转发器将数据发送给管理器节点提供的这组节点。

通过这种方式，转发器保持与群集状态同步，了解已加入或离开群集的所有对等节点，并相应地更新它们的接收对等节点集。

如果是在多站点群集中，每个转发器可以将自己指定为一个站点的成员。在这种情况下，管理器节点只会传输该站点的所有对等节点的列表，并且转发器会限制自己只在该站点间进行负载均衡。请参阅“在多站点群集中使用索引器发现”。

此外，对等节点可以使用**加权负载均衡**，基于每个对等节点相对的磁盘容量，来调整它们发送给该对等节点的数据量。请参阅“使用加权负载均衡”。

注意：如果管理器节点发生故障，转发器将使用最近的可用对等节点列表。然而，在转发器重新启动后不会保留该列表。因此，如果转发器重新启动的同时管理器节点发生故障，则转发器上将没有对等节点列表并将无法转发数据，可能会导致数据丢失。同样地，如果转发器是第一次启动，它必须等待管理器节点返回信息，才能获得对等节点列表。

配置索引器发现

以下是使用索引器发现来设置转发器与对等节点之间的连接的主要步骤：

1. 将对等节点配置为从转发器接收数据。

2. 配置管理器节点启用索引器发现。

3. 配置转发器。

在建立连接之后，您必须在转发器上配置数据导入。请参阅“配置每个转发器的数据导入”。

1. 将对等节点配置为从转发器接收数据

要使某对等节点从转发器接收数据，必须配置该对等节点的**接收端口**。指定接收端口的一种方法是编辑对等节点的 inputs.conf 文件。例如，inputs.conf 中的该设置将接收端口设为 9997：

```
[splunktcp://9997]
disabled = 0
```

更改后重新启动对等节点。

请参阅*《转发数据手册》*中的“启用接收器”。

警告：当使用索引器发现时，每个对等节点只能有一个配置的接收端口。端口可以配置用于 splunktcp 或 splunktcp-ssl，但不能同时用于两者。对于群集中的所有对等节点，您必须使用相同的方法：splunktcp 或 splunktcp-ssl。

您可以通过在所有对等节点上部署相同且单一的 inputs.conf 文件，简化对等节点输入配置。在 inputs.conf 的通用副本中指定的接收端口将代替您在各个对等节点上启用的任何端口。如何在所有对等节点中创建和部署通用 inputs.conf 的详细信息，请参阅更新通用对等节点配置和应用。

当转发到多站点群集时，您可以配置转发器只发送数据到指定站点中的对等节点。请参阅“在多站点群集中使用索引器发现”。

2. 配置管理器节点启用索引器发现

在管理器节点上的 server.conf 中，添加以下段落：

```
[indexer_discovery]
pass4SymmKey = <string>
polling_rate = <integer>
indexerWeightByDiskCapacity = <bool>
```

请注意以下事项：

- pass4SymmKey 属性指定用于群集管理器节点和转发器之间通信的安全密钥。对于所有转发器和管理器节点，该值必须相同。用于 indexer_discovery 的 pass4SymmKey 属性的值应与用于管理器节点和群集节点之间的通信的 pass4SymmKey 属性（此属性设置于 [clustering] 段落之内值不同），详细介绍请参阅“配置安全密钥”。
- polling_rate 属性（可选）提供了一种方法，来调整转发器轮询管理器节点以获得对等节点最新列表的速率。它的值必须是 1 到 10 之间的整数。默认值为 10。请参阅“调整轮询频率”。
- indexerWeightByDiskCapacity 属性（可选）决定了索引器发现是否使用加权负载均衡。默认值为 false。请参阅“使用加权负载均衡”。

3. 配置转发器

a. 配置转发器使用索引器发现

在每个转发器上，添加以下设置到 outputs.conf 文件：

```
[indexer_discovery:<name>]
pass4SymmKey = <string>
master_uri = <uri>

[tcpout:<target_group>]
indexerDiscovery = <name>

[tcpout]
defaultGroup = <target_group>
```

请注意以下事项：

- 在 [indexer_discovery:<name>] 段落中，<name> 参考 <name> 设置（属于 indexerDiscovery 属性，位于 [tcpout:<target_group>] 段落中）。
- pass4SymmKey 属性指定用于管理器节点和转发器之间通信的安全密钥。对于所有转发器和管理器节点，该值必须相同。必须以显式方式为每个转发器设置此值。
- <master_uri> 是管理器节点的 URI 和管理端口。例如："https://10.152.31.202:8089"。
- 在 [tcpout:<target_group>] 段落中，设置 indexerDiscovery 属性，而不是 server 属性（如果您不启用索引器发现，则使用该属性来指定接收对等节点）。通过使用索引器发现，转发器会从管理器节点而不是从 server 属性获得它们的接收对等节点列表。如果同时设置了两个属性，则 indexerDiscovery 优先。

b. 在每个转发器上启用索引器确认

注意：此步骤是确保端到端的数据保真度所必需的。如果部署不要求这一点，则可以跳过此步骤。

要确保群集接收所有传入数据并为其建立索引，必须在每个转发器上启用索引器确认功能。

要配置索引器确认，设置 useACK 属性（位于每个转发器的 outputs.conf 中），与您设置的 indexerDiscovery 属性位于同一段落：

```
[tcpout:<target_group>]
indexerDiscovery = <name>
useACK=true
```

配置索引器确认的详细信息，请参阅*《转发数据手册》*中的“防止传输中的数据丢失”。

示例

在本例中：

- 管理器节点启用索引器发现。
- 管理器节点和转发器共享一个安全密钥。
- 转发器将数据发送给通过对等节点磁盘的总磁盘容量进行加权的对等节点。
- 转发器使用索引器确认来确保端到端数据保真度。

在管理器节点中：server.conf：

```
[indexer_discovery]
pass4SymmKey = my_secret
indexerWeightByDiskCapacity = true
```

在每个转发器的 outputs.conf 中：

```
[indexer_discovery:manager1]
pass4SymmKey = my_secret
master_uri = https://10.152.31.202:8089

[tcpout:group1]
autoLBFrequency = 30
forceTimebasedAutoLB = true
indexerDiscovery = manager1
useACK=true

[tcpout]
defaultGroup = group1
```

在多站点群集中使用索引器发现

在多站点群集化时，通常基于群集节点的位置，将群集划分到站点中。请参阅多站点索引器群集。当将索引器发现与多站点群集化配合使用时，可以配置每个转发器进行站点识别，以便将数据仅转发给单个指定站点上的对等节点。

当将索引器发现与多站点群集化配合使用时，不管您是否希望每个转发器都进行站点识别，必须给所有转发器分配一个 site-id：

- 如果希望某个转发器进行站点识别，则可为群集中某站点的该转发器分配一个 site-id，如 "site1"、"site2" 等。
- 如果您不希望转发器进行站点识别，则可以给它分配特殊 site-id ("site0")。当为转发器分配了 "site0" 时，它会转发数据给群集中所有站点上的对等节点。

为每个转发器分配一个 site-id

要分配一个 site-id，请添加该段落到转发器的 server.conf 文件中：

```
[general]
site = <site-id>
```

请注意以下事项：

- 必须为发送数据到多站点群集的每个转发器分配一个 <site-id>。该值必须为群集中的一个有效站点或是特殊值 "site0"。
- 如果希望转发器只将数据发送给特定站点的对等节点，则为其分配该站点的 ID，如 "site1"。
- 如果您希望转发器向所有站点上的所有对等节点发送数据，则将值设为 "site0"。
- 如果未分配任何 ID，则转发器不会发送数据给任何对等节点。
- 还可以参阅站点值。

配置转发器站点故障转移功能

如果您为转发器分配了特定的站点且该站点出现故障，默认情况下，此转发器不会将故障转移至其他站点。相反，如果分配的站点上没有对等节点可用，转发器即停止转发数据。要避免该问题，您必须配置转发器站点故障转移功能。

要配置转发器站点故障转移功能，请设置管理器节点的 server.conf 文件中的 forwarder_site_failover 属性。

例如：

```
[clustering]
forwarder_site_failover = site1:site2, site2:site3
```

本示例将配置 site1 和 site2 的故障转移站点。如果 site1 故障，则所有配置为向 site1 上的对等节点发送数据的转发器将改为向 site2 上的对等节点发送数据。同样，如果 site2 故障，则所有明确配置为向 site2 上的对等节点发送数据的转发器将改为向 site3 上的对等节点发送数据。

注意：故障转移功能不会从一个站点传递到另一个站点。换言之，前一示例中，如果转发器设为 site1 并且 site1 发生了故障，则转发器将开始向 site2 上的对等节点转发数据。但是，如果 site2 随后发生了故障，则 site1 的转发器不会转发数据到 site3。只有明确设为 site2 的转发器会转发数据到 site3。每个转发器只能有一个故障转移站点。

一旦该站点上的任何对等点返回到群集，转发器将转换为分配的站点。例如，假设管理器节点包含此配置：

```
[clustering]
forwarder_site_failover = site1:site2
```

当 site1 发生故障时，这样就没有对等节点运行在 site1 上，分配给 site1 的转发器开始改向 site2 上的对等节点发送数据。这种故障转移状态会一直持续，直到一个 site1 的对等节点返回到群集。从此时起，分配给 site1 的转发器开始向该对等节点转发数据。它们不再向 site2 上的对等节点转发数据。

使用加权负载均衡

当您启用索引器发现时，转发器总是在一组对等节点间流送传入的数据，使用负载均衡将数据流从一个节点切换到另一个节点。这与没有索引器发现的转发器如何使用负载均衡的方式类似，但有一些关键性的差异。特别是，您可以启用加权负载均衡。

在加权负载均衡中，当转发器对数据进行负载均衡时，转发器会将每个对等节点的磁盘容量考虑在内。例如，使用 400GB 磁盘的对等节点接收到的数据约为使用 200GB 磁盘的对等节点接收数据的两倍。

重要提示：磁盘容量是指对等节点上本地磁盘空间的总量，而不是可用空间的数量。

加权负载均衡如何工作

加权负载均衡的工作方式类似于普通转发器负载均衡。转发器的 autoLBFrequency 属性（位于 outputs.conf 文件中）仍然决定了数据流切换到不同索引器的频率。但是，当转发器选择下一个索引器时，它是基于相对磁盘容量执行此操作的。选择本身是随机的，但是倾向于拥有较大磁盘容量的索引器。

换句话说，转发器采用加权挑选。因此，如果转发器的 autoLBFrequency 设置为 60，那么每六十秒，转发器会将数据流切换到新的索引器。如果负载均衡发生在两个索引器之间，一个有 500GB 磁盘，另一个有 100GB 磁盘，在每个切换时间点，有较大磁盘的索引器被选中的可能性是另一个的五倍。

发送到每个索引器的总流量基于该比例：

```
indexer_disk_capacity/total_disk_capacity_of_indexers_combined
```

有关索引器群集中负载均衡的常规讨论，请参阅负载均衡如何工作。

启用加权负载均衡

indexerWeightByDiskCapacity 属性（位于管理器节点的 server.conf 文件中）控制加权负载均衡：

```
[indexer_discovery]
indexerWeightByDiskCapacity = <bool>
```

请注意以下事项：

- 默认情况下，indexerWeightByDiskCapacity 属性设置为 false。要启用加权负载均衡，您必须将它设为 true。

更改索引器公布的磁盘容量

在一些情况下，您可能希望加权负载均衡对待索引器就好像它所拥有的磁盘容量比它实际上有的更低。您可以使用 advertised_disk_capacity 属性来实现这点。例如，如果您在有 500GB 磁盘的索引器上设置该属性为 50（表示 50%），则进行加权负载均衡时将认为实际磁盘容量为 250GB。

您可以设置 advertised_disk_capacity 属性（位于索引器的 server.conf 文件中）：

```
[clustering]
advertised_disk_capacity = <integer>
```

请注意以下事项：

- advertised_disk_capacity 属性指示在索引器发送容量到管理器节点之前，将应用于其实际磁盘容量的百分比。例如，如果

在有 500GB 磁盘的索引器将其设置为 50，则索引器会告诉管理器节点，它的磁盘容量是 250GB。
- 该值可以在 10 到 100 之间变化。
- 默认值为 100。

调整轮询频率

转发器会按固定的时间间隔轮询管理器节点，以接收对等节点的最新列表。这样，它们会了解可用对等节点集的任何变化，并能相应地修改它们的转发。您可以调整轮询速率。

轮询的频率基于转发器的数量和 `polling_rate` 属性的值（在管理器节点的 `server.conf` 文件中配置）。每个转发器的轮询间隔均遵循该公式：

```
(number_of_forwarders/polling_rate + 30 seconds) * 1000 = polling interval, in milliseconds
```

以下是一些示例：

```
# 100 forwarders, with the default polling_rate of 10
(100/10 + 30) * 1000 = 40,000 ms., or 40 seconds

# 10,000 forwarders, with the default polling_rate of 10
(10000/10 + 30) * 1000 = 1,030,000 ms., or 1030 seconds, or about 17 minutes

# 10,000 forwarders, with the minimum polling_rate of 1
(10000/1 + 30) * 1000 = 10,030,000 ms., or 10,030 seconds, or a bit under three hours
```

要配置 `polling_rate`，请添加该属性到 `[indexer_discovery]` 段落（位于管理器节点的 `server.conf` 中）：

```
[indexer_discovery]
polling_rate = <integer>
```

请注意以下事项：

- `polling_rate` 属性必须是 1 到 10 之间的整数。
- 默认值为 10。

使用 SSL 配置索引器发现

您可以使用 SSL 配置索引器发现。是否使用 SSL 配置的过程是一样的，只是有少量新增和修改：

1. 将对等节点配置为通过 SSL 从转发器接收数据。

2. 配置管理器节点启用索引器发现。

3. 为转发器进行 SSL 配置。

以下步骤仅说明基本配置信息，主要针对配置 SSL 时的差异之处。有关索引器发现配置的完整详细信息，请参阅 配置索引器发现。

1. 将对等节点配置为通过 SSL 从转发器接收数据

编辑每个对等节点的 `inputs.conf` 文件，指定接收端口并配置必需的 SSL 设置：

```
[splunktcp-ssl://9997]
disabled = 0

[SSL]
serverCert = <path to server certificate>
sslPassword = <certificate password>
```

注意：当使用索引器发现时，每个对等节点只能有一个接收端口。对于 SSL，必须将端口配置为只支持 `splunktcp-ssl`。不要配置 `splunktcp` 段落。

另外，确认每个对等节点的 `server.conf` 文件中都设置了 `sslRootCAPath`。

2. 配置管理器节点启用索引器发现

在管理器节点上的 server.conf 中，添加以下段落：

```
[indexer_discovery]
pass4SymmKey = <string>
polling_rate = <integer>
indexerWeightByDiskCapacity = <bool>
```

这和配置非 SSL 设置是一样的。

3. 为转发器进行 SSL 配置

在每个转发器上，添加以下设置到 outputs.conf 文件：

```
[indexer_discovery:<name>]
pass4SymmKey = <string>
master_uri = <uri>

[tcpout:<target_group>]
indexerDiscovery = <name>
useACK = true
clientCert = <path to  client certificate>
sslPassword = <CAcert password>

[tcpout]
defaultGroup = <target_group>
```

另外，确认每个转发器的 server.conf 文件中都设置了 sslRootCAPath。

直接连接转发器与对等节点

以下是使用传统方法将每个转发器直接连接到每个对等节点，来设置转发器与对等节点之间的连接的主要步骤：

1. 将对等节点配置为从转发器接收数据。

2. 将转发器配置为将数据发送到对等节点。

3. 在每个转发器上启用索引器确认。此步骤是确保端到端的数据保真度所必需的。如果部署不要求这一点，则可以跳过此步骤。

在建立连接完成之后，您必须在转发器上配置数据导入。请参阅“配置转发器的数据导入”。

1. 将对等节点配置为从转发器接收数据

要使某对等节点从转发器接收数据，必须配置该对等节点的**接收端口**。有关如何配置接收端口的信息，请参阅*《转发数据》*手册中的“启用接收器”。

指定接收端口的一种方法是编辑对等节点的 inputs.conf 文件。您可以通过在所有对等节点上部署相同且单一的 inputs.conf 文件，简化对等节点输入配置。在 inputs.conf 的通用副本中指定的接收端口将代替您在各个对等节点上启用的任何端口。有关如何在所有对等节点中创建和部署通用 inputs.conf 的详细信息，请参阅“更新通用对等节点配置”。

2. 将转发器配置为将数据发送到对等节点

设置转发器时，应指定其接收对等节点，即提供该节点的 IP 地址和接收器端口号。例如：10.10.10.1:9997。在转发器的 outputs.conf 文件中执行此操作，如*《转发数据》*手册中的“使用 outputs.conf 配置转发器”所述。要指定接收对等节点，请按如下所示设置 server 属性：

```
server=10.10.10.1:9997
```

此处指定的接收端口为步骤 1 中您在对等节点上配置的端口。

要将转发器设置为使用负载均衡，以使数据依次转到多个对等节点，请配置接收节点的负载均衡组。例如，outputs.conf 中的此属性/值对指定了一个由三个对等节点组成的负载均衡组：

```
server=10.10.10.1:9997,10.10.10.2:9997,10.10.10.3:9997
```

要了解有关配置负载均衡的详细信息，请参阅*《转发数据》*手册中的“设置负载均衡”。

注意：还可以通过其他多种方式指定转发器的接收对等节点。例如：

- 可在部署通用转发器期间指定接收对等节点（仅适用于 Windows 通用转发器），如*《通用转发器》*手册中的“通过安装程序安装 Windows 通用转发器”所述。
- 可使用 CLI 命令 `add forward-server` 指定接收器，如*《转发数据》*手册中的“启用接收器”所述。

这两种方法都可以通过基本 `outputs.conf` 文件来实现。无论您使用何种方法来指定接收对等节点，仍需要通过直接编辑基本 `outputs.conf` 文件来启用索引器确认，如下一步中所述。

3. 在每个转发器上启用索引器确认

此步骤是确保端到端的数据保真度所必需的。如果部署不要求这一点，则可以跳过此步骤。

要确保群集接收所有传入数据并为其建立索引，必须在每个转发器上启用索引器确认功能。

警告：某些情况下，索引器确认可能会产生重复事件。要了解此问题以及如何解决问题，请参阅*转发数据*手册中的“防止传输中的数据丢失”。

要配置索引器确认，设置 `useACK` 属性（位于每个转发器的 `outputs.conf` 中）：

```
[tcpout:<peer_target_group>]
useACK=true
```

配置索引器确认的详细信息，请参阅*《转发数据手册》*中的“防止传输中的数据丢失”。

警告：为使索引器确认正常工作，转发器的等待队列必须配置为最佳大小。对于 5.0.4 或更高版本的转发器，系统将自动进行处理。对于较早版本的转发器，遵照*该转发器版本*的“防止传输中的数据丢失”主题的版本说明。具体地说，阅读调整 `maxQueueSize` 设置上的子主题。

示例：具有索引器确认功能的负载均衡转发器

下面是使用负载均衡依次向群集中的三个对等节点发送数据的转发器的一个示例 `outputs.conf` 配置。它假定每个对等节点之前已被配置为对其接收端口使用 9997：

```
[tcpout]
defaultGroup=my_LB_peers

[tcpout:my_LB_peers]
autoLBFrequency=40
server=10.10.10.1:9997,10.10.10.2:9997,10.10.10.3:9997
useACK=true
```

该转发器首先将数据发送到 `server` 属性中列出的其中一个对等节点。40 秒后，它会切换到另一个对等节点，依此类推。如果转发器没有接收到来自当前接收节点的确认，它就会重新发送数据，但此时将数据发送到下一个可用的节点。

配置索引器群集

索引器群集配置概述

要配置索引器群集，您需要配置单个节点。请执行如下两种类型的配置：

- 配置群集自身的行为。
- 配置群集的索引和搜索行为。

本章特别提供配置群集行为的方法概述。

配置每种节点类型

每种节点类型的设置处理群集的不同方面：

- **管理器节点。**配置整个群集的行为。
- **对等节点。**配置单个对等节点和群集索引行为。
- **搜索头。**在索引器群集中配置单个搜索头和搜索行为。

请参阅关于特定节点类型的章节获得关于配置每种节点类型的信息。例如，“配置对等节点”这一章包括了一些关于配置对等群集节点设置的主题，以及介绍如何配置节点使用的索引的其他主题。

配置群集行为的方法

每个节点的初始配置发生在部署期间。如果您需要更改群集节点的后部署配置，您有如下选择：

- 您可在 Splunk Web 中的节点仪表板中编辑配置，如同“使用仪表板配置索引器群集”中所述。
- 可以直接编辑 [clustering] 段落（位于节点的 server.conf 文件中）。有关详细信息，请参阅“使用 server.conf 配置索引器群集”。要配置一些高级的设置，必须编辑此文件。
- 您可以使用 CLI。有关详细信息，请参阅“使用 CLI 配置和管理索引器群集”。

使用仪表板配置索引器群集

通过索引器群集节点的仪表板配置其节点：

1. 单击节点上的 Splunk Web 右上角的**设置**。

2. 在**分布式环境**组中，单击**索引器群集化**。

3. 在仪表板的右上角选择**编辑**按钮。

关于每种节点类型设置的信息，请参阅：

- “使用仪表板配置管理器节点”
- “使用仪表板配置对等节点”
- “使用仪表板配置搜索头”

使用 server.conf 配置索引器群集

在阅读本主题之前，请参阅*《管理员手册》*中的“关于配置文件”及其后续主题。那些主题介绍了 Splunk Enterprise 如何使用配置文件。

索引器群集设置驻留在 server.conf 文件中，位于 $SPLUNK_HOME/etc/system/local/。当您通过 Splunk Web 或 CLI 部署群集节点时，节点将设置保存到该文件中。您也可直接编辑 server.conf 文件，可以进行初始化部署，也可稍后更改设置。

用于控制索引器群集化的主要 server.conf 段落是 [clustering]。除了对应于 Splunk Web 中的设置的基本属性以外，server.conf 提供一些高级设置，可控制群集节点之间的通信。除非 Splunk 支持建议，请不要更改那些设置。

本主题讨论一些对所有节点类型都普遍存在的问题。

配置各种节点类型

关于每种节点类型的特别说明，请参阅：

- “使用 server.conf 配置管理器节点”。
- “使用 server.conf 配置对等节点”。
- “使用 server.conf 配置搜索头”。

有关所有群集化属性（包括高级属性）的详细信息，请阅读 server.conf 规范。

关于多站点群集的配置，也请阅读“使用 server.conf 配置多站点索引器群集”。

配置安全密钥

通过设置 pass4SymmKey 属性来配置验证管理器节点、对等节点与搜索头之间通信的安全密钥。您必须对所有群集节点使用相同的密钥值。

部署群集时请设置 pass4SymmKey。有关如何在管理器节点上设置密钥的详细信息，请参阅“启用索引器群集管理器节点”。您也可以在启用对等节点和搜索头时进行设置。

如果直接在 server.conf 中设置密钥，您必须将其设置在索引器群集化的 [clustering] 段落内。

重要提示：将密钥副本保存在安全的地方。一旦开始运行一个实例，安全密钥就从明文变为加密形式，无法再从 server.conf 中恢复。如果之后您想添加一个新节点，您将需要使用明文形式来设置密钥。

有关为合并搜索头群集和索引器群集设置安全密钥的信息，请参阅*分布式搜索*中的“集成搜索头群集和索引器群集”。

修改 server.conf 后重新启动

在您首次配置实例为群集节点后，则需要重新启动它以使更改生效。

如果之后您要更改配置，您可能不需要重启实例，这取决于更改的类型。尽可能地避免重启对等节点。重启对等节点集会导致长时间大量的数据桶修复。

初始配置

将实例初始配置为群集节点之后，需要重新启动所有节点（管理器节点、对等节点和搜索头）以使更改生效。您可以通过对每个节点调用 CLI restart 命令来实现此目的：

```
$SPLUNK_HOME/bin/splunk restart
```

首次启动管理器节点时，它会阻止在对等节点上建立索引，直到您启用并重新启动个数等于复制因子的对等节点为止。当等待对等节点加入群集时不要重启管理器节点。如果重启了主节点，对等节点将需要再次重启。

重要提示：虽然最初启用 Splunk 实例作为群集对等节点时可以使用 CLI restart 命令，但以后重新启动时不要使用此命令。复制开始后，restart 命令与索引复制不兼容。有关更多信息（包括安全重新启动方法的讨论），请参阅“重新启动单个对等节点”。

后续配置更改

如果您更改了 server.conf 文件中的下列属性，不必重新启动节点。

在对等节点上：

- manager_uri
- notify_scan_period

在搜索头上：

- manager_uri

在管理器节点上：

- quiet_period
- heartbeat_timeout
- restart_timeout
- max_peer_build_load
- max_peer_rep_load
- cluster_label
- access_logging_for_heartbeats
- use_batch_mask_changes
- percent_peers_to_restart

- summary_replication

所有其他群集相关配置的更改都需要重新启动。

使用 CLI 配置和管理索引器群集

您可以使用 CLI 执行广泛的索引器群集活动，包括：

- 配置群集节点
- 查看群集信息
- 管理群集

一些群集化命令仅对特定节点类型可用，例如管理器节点。

本主题讨论对所有节点类型都普遍存在的问题。

配置群集节点

您可以使用 CLI 启用任何群集节点类型或稍后更改它们的配置：

- 要启用或编辑管理器节点，请参阅“使用 CLI 配置管理器节点”。
- 要启用或编辑对等节点，请参阅“使用 CLI 配置对等节点”。
- 要启用或编辑搜索头，请参阅“使用 CLI 配置搜索头”。

关于特定命令行选项的详细信息，请阅读“使用 server.conf 配置索引器群集”。

关于多站点群集配置，也请阅读“使用 CLI 配置多站点索引器群集”。

指定安全密钥

启用每个群集节点时通过附加 -secret 标记为群集指定一个安全密钥。例如，您可以在配置对等节点时指定安全密钥：

```
splunk edit cluster-config -mode peer -manager_uri https://10.160.31.200:8089 -replication_port 9887 -secret your_key
```

安全密钥验证管理器节点与对等节点以及搜索头之间通信。必须指定密钥，而且所有群集节点的密钥必须相同。

-secret 标记编辑 server.conf 的 [clustering] 段落中的 pass4SymmKey 设置。

查看群集信息

有许多 splunk list 命令返回不同类型的群集信息。例如，要获得有关群集中的每个对等节点的详细信息，请在管理器节点上运行以下命令：

```
splunk list cluster-peers
```

要获得有关群集配置的信息，请从任何节点运行以下命令：

```
splunk list cluster-config
```

请参阅“CLI 群集化帮助”，以了解完整的 splunk list 命令集。

管理群集

还可以使用 CLI 对群集执行许多不同的操作。有关这些操作的说明，请参见相应操作的主题：

- 使用 splunk offline 命令使对等节点脱机。
- 使用 splunk apply cluster-bundle 命令更新通用对等节点配置。
- 使用 splunk rolling-restart cluster-peers 命令重新启动所有群集对等节点。
- 使用 splunk enable maintenance-mode 命令启用维护模式。
- 使用 splunk remove excess-buckets 命令删除过多的数据桶副本。
- 配置多群集搜索。

获取 CLI 命令相关帮助

CLI 为其命令提供了联机帮助。要获得有关一组完整群集化命令的常规帮助，请转到 $SPLUNKHOME/bin 并键入：

```
splunk help cluster
```

要获得特定命令的相关帮助，请指定命令名称。例如：

```
splunk help list cluster-config
```

有关 CLI 的一般信息，请参阅*《管理员手册》*中的“使用命令行界面（CLI）管理 Splunk Enterprise”一章，或者键入：

```
splunk help
```

配置管理器节点

管理器节点配置概述

在启用管理器节点时已对其进行初始配置，如“启用管理器节点”中所述。这通常是管理器节点所需的全部配置。

更改配置

如果您需要编辑配置，有如下选择：

- 从 Splunk Web 中的管理器节点仪表板上编辑配置。请参阅“使用仪表板配置管理器节点”。
- 可以直接编辑 [clustering] 段落（位于管理器节点的 server.conf 文件中）。要配置一些高级的设置，必须编辑此文件。请参阅“使用 server.conf 配置管理器节点”。
- 您可以使用 CLI。请参阅“使用 CLI 配置管理器节点”。

在更改管理器节点的配置之后，需要重新启动管理器节点以使更改生效。

重要提示：管理器节点的唯一功能是管理其他群集节点。不要使用主节点来为外部数据建立索引或搜索群集。

需要注意的更改

当更改如下设置时要小心：

- **复制因子和搜索因子。**不建议在索引器群集包含大量数据之后增加这些设置。这样会启动大量的数据桶活动，当数据桶副本产生时或变成可搜索副本时，群集的性能会受到不利影响。
- **检测信号超时。**不要更改 heartbeat_timeout 属性的默认值 60（秒），除非 Splunk 支持要求您这样做。特别是，不要降低这个值。这样会让节点超载。

配置备用管理器节点

为预防管理器节点故障，配置备用管理器节点。这样在当前管理器节点发生故障时可以接管。请参阅“在索引器群集中替换管理器节点”。

配置多站点管理器节点

多站点管理器节点与基本的、单个站点群集相比，有一些配置上的差异和附加项。请参阅“使用 server.conf 配置多站点索引器群集”。

使用仪表板配置管理器节点

您可通过仪表板编辑现有管理器节点的配置：

1. 在 Splunk Web 中单击**设置 > 索引器群集化**。
   显示“管理器节点”仪表板。
2. 在仪表板的右上角选择**编辑**按钮。
   编辑按钮有如下几个选项：

- **节点类型**。更改实例的节点类型。**警告：**您不太可能会希望更改已启用群集中的节点的节点类型。进行更改之前，仔细考虑后果。
- **管理器节点配置**。更改这些管理器节点设置：
  - **复制因子**。更改群集的复制因子。**警告：**不建议群集包含大量数据后增加复制因子。这样做将启动大量数据桶活动，从而在创建数据桶副本时使群集的性能受到负面影响。
  - **搜索因子**。更改群集的搜索因子。**警告：**不建议群集包含大量数据后增加搜索因子。这样做将启动大量数据桶活动，从而在使数据桶副本可搜索时使群集的性能受到负面影响。
  - **安全密钥**。更改安全密钥。如果您还为群集中的所有其他节点更改，则仅更改安全密钥。群集中的所有实例的密钥必须相同。
  - **群集标签**。为群集贴标签。标签对于识别监视控制台中的群集很有用。请参阅*《确保 Splunk Enterprise 安全手册》*中的“设置群集标签”。

  多站点群集的管理器节点配置选项已禁用。

- **配置软件包操作。**单击**推送**以将管理器节点的配置软件包分发到对等节点。还可在不应用软件包的情况下验证软件包和

检查重新启动，或者回滚至之前的软件包。请参阅“更新通用对等节点配置和应用”。
- **数据重新平衡**。重新平衡数据桶，这样每个对等节点都将获得数量大概一致的数据桶副本。请参阅“重新平衡索引器群集”。
- **禁用索引器群集化**。从群集删除本节点。**警告**：如果您从群集删除管理器节点，则整个群集将最终发生故障。

有关首次使用此仪表板启用管理器节点的信息，请参阅“启用索引器群集管理器节点”。

关于使用此仪表板查看群集状态的信息，请参阅“查看管理器节点仪表板”。

使用 server.conf 配置管理器节点

前提条件

在阅读本主题之前，请参阅“使用 server.conf 配置索引器群集”。它介绍了对于所有群集节点类型都普遍存在的配置问题。

启用管理器节点

下面的示例显示当启用一个管理器节点时要配置的基本设置。除非另有说明，否则该设置是必需的。配置属性与 Splunk Web 中的**启用群集化**页面上的字段相对应。

```
[clustering]
mode = manager
replication_factor = 4
search_factor = 3
pass4SymmKey = whatever
cluster_label = cluster1
```

该示例指定了以下内容：

- 实例为群集管理器节点。
- 群集的复制因子为 4。
- 群集的搜索因子为 3。
- 安全密钥为 "whatever"。群集中的所有节点均使用相同的安全密钥。请参阅“配置安全密钥”。
- 群集标签为 "cluster1"。可选的群集标签对于识别监视控制台中的群集很有用。请参阅*《监视 Splunk Enterprise 手册》*中的“设置群集标签”。只在管理器节点上设置此属性。

首次启动管理器节点时，它将阻止在对等节点上建立索引，直到您启用并重新启动了个数等于复制因子的全部对等节点为止。当等待对等节点加入群集时不要重启管理器节点。如果重启了主节点，对等节点将需要再次重启。

在 Splunk Web 中启用管理器节点时，所生成的 server.conf 段落将仅包含使用非默认值的属性。例如，如果您接受默认复制因子为 3，而没有输入新值，那么生成的段落将不包括 replication_factor 属性。

编辑管理器节点设置

如有必要，您可以稍后更改这些设置。例如，要更改群集的安全密钥，可在每个节点上编辑 pass4SymmKey 的值。

关于所有群集属性的详细信息，包括一些很少需要编辑的高级属性，请阅读 server.conf 规范文件。

使用 CLI 配置管理器节点

首先阅读

在阅读本主题前，请参阅：

- “使用 CLI 配置和管理索引器群集”。本主题介绍使用 CLI 进行基本的索引器群集配置。它提供关于所有群集节点类型的普遍问题的详细信息。

启用管理器节点

下面的示例显示当启用一个管理器节点时通常要配置的基本设置。配置属性与 Splunk Web 中的**启用群集化**页面上的字段相对应。

```
splunk edit cluster-config -mode manager -replication_factor 4 -search_factor 3 -secret your_key -cluster_label cluster1

splunk restart
```

-secret 标记编辑 server.conf 的 [clustering] 段落中的 pass4SymmKey 设置。

首次启动管理器节点时，它将阻止在对等节点上建立索引，直到您启用并重新启动了个数等于复制因子的全部对等节点为止。当等待对等节点加入群集时不要重启管理器节点。如果重启了主节点，对等节点将需要再次重启。

编辑管理器节点设置

您还可以稍后使用 CLI 来编辑配置。使用 splunk edit cluster-config 命令，此命令和最初启用管理器节点所用的命令相同。

有关可配置设置的列表，请参阅 CLI 群集化帮助，以及 server.conf 规范文件。

警告：切勿在管理器节点上增加复制因子或搜索因子

尽管可以更改复制因子和搜索因子的设置，但是在群集包含大量数据之后增加这两个因子中的任何一个都不是明智之举。这样做将启动大量数据桶活动，从而在创建数据桶副本或使数据桶副本成为可搜索副本时使群集的性能受到负面影响。

在索引器群集中替换管理器节点

您可能因为这些原因需要替换管理器节点：

- 节点故障。
- 必须移动管理器节点到不同的计算机或站点。

虽然目前没有管理器节点故障转移操作，但您可以通过配置备用管理器节点来为管理器节点故障准备索引器群集。如果主要管理器节点故障，您可以立即启动备用管理器节点。您可以使用同样的方法有意地替换管理器节点。

本主题介绍替换管理器节点的关键步骤：

1. 备份替换管理器节点需要的文件。

   这是一个预备步骤。您必须在管理器节点故障前执行此步骤，否则只能离开系统。

2. 确保对等节点和搜索头节点能找到新的管理器节点。
3. 更换管理器。

在多站点群集的情况下，您也必须为包括管理器节点的站点可能出现的故障做准备。请参阅“处理管理器站点故障”。

备份替换管理器节点需要的文件

您必须备份多个文件和目录，这样您后续才将其复制到替换管理器节点：

- 管理器节点的 server.conf 文件，这是管理器节点群集设置的存储位置。您每次更改管理器节点的群集配置时，都必须备份此文件。
- 管理器节点的 $SPLUNK_HOME/etc/master-apps 目录，这是通用对等节点配置的存储位置，如“更新群集对等节点配置”所述。每次更新推送到对等节点的一组内容时，都必须备份此目录。
- 管理器的 $SPLUNK_HOME/var/run/splunk/cluster/remote-bundle/ 目录，其中包含推送到对等节点的实际配置软件包。每次将新内容推送到对等节点时，都必须备份此目录。

如果 $SPLUNK_HOME/var/run/splunk/cluster/remote-bundle/ 目录包含大量旧软件包，则可以选择仅备份与当前及之前处于活跃状态的软件包相关联的文件。查找以 .bundle_active 和 .bundle_previousActive 结尾的两个文件。这些文件中的每个文件都有一个关联的目录和一个文件，标有软件包 ID。您必须总共备份所有六个文件/目录。

例如，如果目录包含文件 42af6d880c6a1d43e935e8d8a0062089-1571637961.bundle_active，则也会包含文件 42af6d880c6a1d43e935e8d8a0062089-1571637961.bundle 和目录 42af6d880c6a1d43e935e8d8a0062089-1571637961。要备份活跃软件包，必须备份这两个文件和目录。同样，要备份之前处于活跃状态的软件包，必须备份以 .bundle_previousActive 结尾的文件，以及目录和其他具有相同 ID 的文件。

除上述文件和目录外，请备份您在管理器节点上自定义的所有其他配置文件，如 inputs.conf 和 web.conf 等。

在准备替换管理器时，必须仅复制这些文件和目录。您不用复制或处理群集的动态状态。群集对等节点作为一个组保存有关群集动态状态的所有信息，例如所有数据桶副本的状态。举例来说，当停机的管理器节点又回到群集或备用管理器节点替代了停机的管理器节点时，群集对等节点会根据需要将此信息传达给管理器节点，然后管理器节点会使用此信息重新构建群集动态状态地图。

确保对等节点和搜索头节点能找到新的管理器节点

您可以在两种方法之间选择，以确保对等节点和搜索头可以找到替换实例并将其识别为管理器节点：

- **替换管理器节点使用与管理器节点相同的 IP 地址和管理端口。**为确保替换主节点使用相同的 IP 地址，您必须采用基于 DNS 的故障转移、负载均衡器或一些其他技术。管理端口在安装过程中设置，但可以通过编辑 web.conf 进行更改。
- **替换管理器节点不使用与管理器节点相同的 IP 地址和管理端口。**在这种情况下，在启用新管理器节点后，您必须更新所有对等节点和搜索头上的 manager 设置，以指向新管理器节点的 IP 地址和管理端口。

任何一种方法都不需要重新启动对等节点或搜索头节点。

更换管理器

前提条件

您必须所需文件和目录集的最新备份，如“备份替换管理器节点需要的文件”中所述。

步骤

如果想跳过步骤 3 和步骤 5，可以在步骤 4 中更换属于替换管理器节点的 [general] 和 [clustering] 段落，而不是复制整个 server.conf 文件。

1. 停止旧的管理器节点，如果计划替换它的话。如果是因管理器节点故障进行替换，那么这一步已经完成了。
2. 安装、启动并停止一个新的 Splunk Enterprise 实例。或者，也可以重新使用无其他用途的现有实例。这将是替换管理器节点。
3. 复制 sslPassword 设置，将其从替换管理器节点的 server.conf 文件中移到一个临时位置。

   在版本 6.5 中，已弃用 sslKeysfilePassword 属性，由 sslPassword 属性所取代。如果 server.conf 文件正在使用 sslKeysfilePassword，则复制该设置来替代它。

4. 复制旧管理器节点的 server.conf 文件的备份到替换管理器节点。
5. 删除 sslPassword 设置（位于复制的 server.conf 中），并用您在步骤 3 中保存的设置版本替代它。
6. 删除复制的 server.conf 中 pass4symmkey 的加密值，并将其替换为纯文本值。请参阅“配置安全密钥”。
7. 复制旧管理器节点的 $SPLUNK_HOME/etc/master-apps 目录的备份到替换管理器节点。
8. 复制旧管理器节点的 $SPLUNK_HOME/var/run/splunk/cluster/remote-bundle/ 目录的备份到新管理器节点。
9. 启动替换管理器节点。
10. 确保通过如“确保对等节点和搜索头节点能找到新的管理器节点”所述的一种方法，使对等节点和搜索头节点指向新的管理器节点。

有关管理器节点出现故障导致的后果的信息，请阅读“管理器节点故障时的情况”。

配置对等节点

对等节点配置概述

配置对等节点分为如下两类：

- 配置基本的索引器群集设置，如管理器节点 URI 和复制端口。
- 配置输入、索引及相关设置。这包含了对对等节点的应用部署。

初始配置

大部分对等节点群集配置都是在初始部署期间进行：

1. 启用对等节点时，指定其群集设置，如管理器节点以及其用来接收复制的数据的端口。请参阅“启用对等节点”。

2. 在启用一组对等节点后，如果需要，配置这些节点的索引。请参阅“在索引器群集中配置对等节点索引”。

3. 最后，配置其输入（通常使用转发器）。请参阅“使用转发器将数据导入索引器群集”。

这些是配置对等节点的主要步骤。与任何索引器一样，您可能还需要在稍后更新配置。

更改群集配置

更改群集节点配置有两个主要原因：

- **重定向对等节点到另一个管理器节点。**当管理器节点故障时这样做非常有用，但是您得有一个备用管理器节点准备好接管。关于备用管理器节点的信息，请参阅“在索引器群集中替换管理器节点”。
- **更改该群集的对等节点的安全密钥。**如果您还为群集中的所有其他节点更改，则仅更改安全密钥。群集中的所有实例的密钥必须相同。

要编辑群集设置，单独更改每个对等节点，可以使用以下方法之一：

- 从 Splunk Web 中的对等节点仪表板上编辑配置。请参阅“使用仪表板配置对等节点”。
- 编辑对等节点的 server.conf 文件。有关详细信息，请参阅“使用 server.conf 配置对等节点”。
- 使用 CLI。有关详细信息，请参阅“使用 CLI 配置对等节点”。

关于配置多站点对等节点时的附加项和不同点，请参阅“使用 server.conf 配置多站点索引器群集”。

配置索引和相关的行为

indexes.conf 中的一组索引段落必须在所有对等节点中都相同，极少的一些例外情况在“按对等节点管理配置”中有介绍。重要的还有：各对等节点上的索引时间处理也务必保持一致。要使群集正确复制数据和处理节点故障，对等节点必须共享相同的索引功能，如果某些关键文件在各对等节点之间不同，这些对等节点将无法做到这一点。

最佳做法是应当将对等节点视为可以互换，因此应在所有对等节点中维护相同的配置文件版本和应用版本。至少，下面这些文件应该一样：

- indexes.conf
- props.conf
- transforms.conf

为了确保对等节点共享一组通用的配置文件和应用，可将文件和应用放在管理器节点上，然后使用**配置软件包**的方法将其分布到这组对等节点上（只需一步操作）。

下面这些主题介绍如何在一组对等节点之间保持相同的配置：

- 在所有对等节点中管理通用配置
- 在所有对等节点中管理应用部署
- 在索引器群集中配置对等节点索引
- 更新通用对等节点配置和应用

管理单个对等节点配置

为了测试或其他目的，您可能偶尔需要按对等节点来处理一些配置。但正常情况下，最好在所有对等节点之间使用相同的配

置，以使对等节点可以互换。

关于单个对等节点配置的信息，请参阅“按对等节点管理配置”。

使用仪表板配置对等节点

您可通过仪表板编辑现有对等节点的配置。访问仪表板：

1. 单击 Splunk Web 右上角的**设置**。

2. 在**分布式环境**组中，单击**索引器群集化**。

3. 在仪表板的右上角选择**编辑**按钮。

编辑按钮提供了一些会影响到配置的选项。

注意：编辑按钮对多站点群集来说是禁用的。

有关首次使用此仪表板启用对等节点的信息，请参阅“启用对等节点”。

关于使用此仪表板查看群集状态的信息，请参阅“查看对等节点仪表板”。

更改群集配置

要更改对等节点的配置，则选择**配置**选项：

- 要更改管理器节点，则编辑**管理器 URI** 字段。
- 要更改复制端口，则编辑**对等节点复制端口**字段。
- 要更改安全密钥，则编辑**安全密钥**字段。

从群集删除对等节点

要从群集中删除对等节点，则选择**禁用索引器群集化**选项。

其他编辑

编辑按钮有一个其他选项：**节点类型**。

您不太可能会希望更改已启用群集中的节点的节点类型。进行更改之前，仔细考虑后果。

使用 server.conf 配置对等节点

前提条件

在阅读本主题之前，请参阅“使用 server.conf 配置索引器群集”。它介绍了对于所有群集节点类型都普遍存在的配置问题。

启用对等节点

下面的示例显示了当启用一个对等节点时您必须要配置的基本设置。这些示例中的配置属性与 Splunk Web 中的**启用群集化**页面上的字段相对应。

```
[replication_port://9887]

[clustering]
manager_uri = https://10.152.31.202:8089
mode = peer
pass4SymmKey = whatever
```

该示例指定了以下内容：

- 此对等节点将使用端口 9887 来侦听从其他对等节点流入的复制的数据。您可指定任意可用但未使用的端口作为复制端口。请勿重复使用管理或接收端口。
- 此对等节点的管理器节点驻留在 10.152.31.202:8089。
- 实例为对等节点。
- 安全密钥为 "whatever"。

必须重新启动实例使设置生效。

编辑对等节点设置

如有必要，您可以稍后更改这些设置。例如，要更改群集的安全密钥，可在每个节点上更改 pass4SymmKey 的值。

使用 CLI 配置对等节点

首先阅读

在阅读本主题前，请参阅：

- 使用 CLI 配置和管理索引器群集。本主题介绍使用 CLI 进行基本的群集配置。它提供关于所有索引器群集节点类型的普遍问题的详细信息。

启用对等节点

下面的示例显示当启用一个对等节点时通常要配置的基本设置。配置属性与 Splunk Web 中的**启用群集化**页面上的字段相对应。

要启用一个实例作为对等节点，请将 mode 设置为 "peer"。您需要指定 manager_uri（用于指定群集的管理器节点），和 replication_port。此外，您必须指定用于整个群集的安全密钥 (secret)：

```
splunk edit cluster-config -mode peer -manager_uri https://10.160.31.200:8089 -replication_port 9887 -secret your_key

splunk restart
```

-secret 标记编辑 server.conf 的 [clustering] 段落中的 pass4SymmKey 设置。

编辑对等节点设置

您还可以稍后使用 CLI 来编辑配置。使用 splunk edit cluster-config 命令，此命令和最初启用对等节点所用的命令相同。

有关可配置设置的列表，请参阅 CLI 群集化帮助，以及 server.conf 规范文件。

在所有对等节点中管理通用配置

您应该尝试在索引器群集中的所有对等节点之间维护一组通用配置文件（包括应用）。通过使对等节点基本上可以互换，增强了高可用性。此外，某些配置必须相同，以便所有的对等节点以相同的方式索引数据。

通过**配置软件包**方法，管理器节点以单个操作将文件和应用分发到所有对等节点。您必须使用该方法来管理通用配置。请参阅“更新通用对等节点配置和应用”。

在所有对等节点之间需要保持一致的配置文件

强烈建议您在所有对等节点之间分发这些文件的相同版本：

- indexes.conf. 所有对等节点共享相同的群集索引集，这一点至关重要。
- props.conf 和 transforms.conf。在索引数据时，所有对等节点必须使用相同的一组规则。

除了这三个关键文件以外，您还可以通过使其他配置文件在所有对等节点之间保持一致的版本，极大地简化群集管理。例如，如果对等节点能够共享一组输入，就可以在所有对等节点之间维护单个的 inputs.conf 文件。

因为应用常常包含那些配置文件的各种版本，所以您通常应该将相同的应用集分发给所有对等节点，而不是将它们分别安装在单个对等节点上。请参阅“在所有对等节点中管理应用部署”。

注意：在有限的情况下（例如，要执行本地测试或监视），您可能只希望向一个对等节点而不是其他的对等节点添加索引。只要您在配置索引时多加注意并清楚后果，您就可以通过创建单个对等节点的 indexes.conf 来实现此目的。这种索引中的数据将不会被复制。单个对等节点的 indexes.conf 可以补充（但不能取代）所有对等节点获得的通用文件版本。如果需要，您可以按同样方法维护单个对等节点的应用。请参阅“将索引添加到单个对等节点”。

分发配置文件到所有对等节点

要在对等节点间分发配置：

1. 如果分发任何 indexes.conf 文件，请配置它们以使它们支持索引复制。请参阅“在索引器群集中配置对等节点索引”。

2. 将文件放在管理器节点上的 $SPLUNK_HOME/etc/master-apps 目录中。该位置上的一组子目录组成配置软件包。

3. 使用 Splunk Web 或 CLI 将配置软件包分发给对等节点。

有关这些步骤的详细信息，请参阅“更新通用对等节点配置和应用”。

与独立索引器相比的对等节点的配置管理

配置软件包方法是在一组对等节点之间管理通用配置和应用部署的唯一支持的方法。它可确保所有对等节点都使用这些文件的相同版本。

注意对等节点配置文件的管理方式与独立索引器的配置之间存在这些重要差异：

- 切勿在单个对等节点上进行配置更改，因为这将修改需要在整个群集内使用的配置。例如，不要使用 Splunk Web 或 CLI 配置索引设置。
- 不要直接在对等节点上编辑群集范围的配置文件，如 indexes.conf。而是要编辑管理器节点上的文件并通过配置软件包的方法去分发。
- 不要使用部署服务器或任何第三方部署工具（比如 Puppet 或 CFEngine）管理对等节点之间的通用配置文件。应改为使用配置软件包方法。

通过配置软件包方法分发更新时，管理器节点会精心安排分发，确保所有对等节点使用相同的一组配置，包括相同的一组群集索引。

如果您不管所有的建议，仍选择使用其他分发方法而非配置软件包方法，则必须至少确保任何新群集索引的设置都已成功分发到所有对等节点，同时确保在开始将数据发送到新索引之前重新加载所有对等节点。

注意：尽管无法使用部署服务器直接分发应用到对等节点，但是您可将它用于分发应用到管理器节点的配置软件包位置。一旦应用在位置中，管理器节点可在随后通过配置软件包方法分发它们到对等节点。请参阅“使用部署服务器将应用分布到管理器节点”。

在所有对等节点中管理应用部署

在阅读本主题之前，请参阅“在所有对等节点中管理通用配置”。应用部署只是该主题介绍的配置文件部署的一种特殊情况。

您必须使用管理器节点将应用部署到对等节点上。不要使用部署服务器或任何第三方部署工具（比如 Puppet 或 CFEngine）。

将应用分发到对等节点

要在对等节点间分发应用：

1. 检查应用的 indexes.conf 文件。对于在应用特定的 indexes.conf 文件中定义的每个索引，设置 repFactor=auto，以便能够在所有对等节点之间复制索引。请参阅“indexes.conf repFactor 属性”。
2. 将应用放在管理器节点上的 $SPLUNK_HOME/etc/master-apps 目录中。该位置上的一组子目录组成**配置软件包**。
3. 使用 Splunk Web 或 CLI 将配置软件包分发给对等节点。

   有关这些步骤中每一步的详细信息，请参阅“更新通用对等节点配置和应用”。

一旦分发应用到一组对等节点，您将使用 Splunk Web 按正常方式启动每个对等节点。请参阅*《管理员手册》*中的“认识 Splunk 应用”章节。

要访问应用时，您可以从搜索头而不是单个对等节点访问。因此，您还必须在搜索头上安装应用。在搜索头上，将应用放到应用的常规位置，即 $SPLUNK_HOME/etc/apps 目录下。根据应用的特定说明，以常规方式安装应用。

删除对等节点中的应用

要删除您之前分发给对等节点的应用，请从配置软件包中删除其目录。当您下次推送软件包的时候，该应用会从所有对等节点中删除。

在索引器群集中配置对等节点索引

可通过编辑 indexes.conf 文件配置索引。此文件确定了索引器的索引集，以及其**数据桶**的大小和属性。由于群集中的所有对等节点必须使用相同的一组索引（下文介绍的几种有限制的应用除外），所以通常所有对等节点上的 indexes.conf 文件都应相同。

群集对等节点使用对等节点特定默认的 indexes.conf 文件进行部署，此文件可处理基本群集需求。如果要添加索引或更改数据桶行为，可在管理器节点上的一个特殊位置编辑新的 indexes.conf 文件，然后将此文件同时分布到所有对等节点。

重要提示：不能使用 Splunk Web 或 CLI 来配置对等节点上的索引设置。必须直接编辑 indexes.conf。

所有对等节点必须使用相同的一组 indexes.conf 文件

通常，群集中的所有对等节点上的一组 indexes.conf 文件均应相同。特别是，所有对等节点必须使用同一组群集索引。这对于索引复制的正常工作非常重要。（另一方面，管理器节点有其自己的单独 indexes.conf 文件，因为它只为自己的内部数据建立索引。）此限制有一个有限的例外情况，下文将加以介绍。

第一次创建群集时，管理器节点会将特别的默认 indexes.conf 文件分布到各个对等节点。本版本补充所有索引器获得的标准默认 indexes.conf。对等节点特定默认 indexes.conf 启动复制 main 索引和内部索引（如 _audit 和 _internal）。

视系统而定，您可能还需要编辑 indexes.conf 并将修改后的文件分发到各对等节点，以便根据数据桶属性调整附加的索引或更改。为了确保所有对等节点使用相同的 indexes.conf，必须在一个过程中使用管理器节点将该文件分发到所有对等节点。此过程称为**配置软件包**方法，在“更新通用对等节点配置”中有所介绍。

还必须使用配置软件包方法在所有对等节点间分发应用。这些应用可能包含自己的 indexes.conf 文件，这些文件与您可能也会分发到对等节点的非应用版本文件一起适当划分层级。关于应用分发的信息，请阅读“在所有对等节点中管理应用部署”。

注意：在受限情况下（例如，要执行本地测试或监视），您可以仅为单个对等节点创建 indexes.conf。这种索引不会被复制。单个对等节点的 indexes.conf 可以补充（但不能取代）所有对等节点获得的通用文件版本。有关详细信息，请参阅“将索引添加到单个对等节点”。

为对等节点配置一组索引

在一组对等节点之间配置索引分为两个步骤：

1. 在管理器节点上编辑一个通用 indexes.conf 文件。

2. 使用管理器节点在该组对等节点之间分发该文件。

下面介绍这两个步骤。

1. 编辑 indexes.conf

有关配置 indexes.conf 的详细信息，请阅读本手册中的“管理索引”和“管理索引存储”章节中的主题。有关所有 indexes.conf 属性的列表，请参阅*《管理员手册》*中的 indexes.conf 规范文件。

其中大多数内容，编辑群集对等节点 indexes.conf 的方法与编辑任何索引器的方法相同。但是，有几点区别需要注意。

indexes.conf repFactor 属性

添加新索引段落时，必须将 repFactor 属性设置为 "auto"。这会使索引的数据复制到群集中的其他对等节点。例如：

```
[idx1]
repFactor = auto
homePath = $SPLUNK_DB/$_index_name/db
coldPath = $SPLUNK_DB/$_index_name/colddb
thawedPath = <path for thawed buckets>
...
```

注意：repFactor 默认设置为 0，表示该索引不能复制。对于群集索引，则必须将其设为 "auto"。

repFactor 的有效值仅为 0 和 "auto"。

将 repFactor 从 "auto" 重置为 0 将停止进一步的复制，但它不会自动删除已复制的数据桶副本。此外，在有多个副本的数据桶之间进行搜索将返回重复的事件。要释放相关的磁盘空间，并消除重复事件的可能性，您必须手动删除多余的副本。

使用正斜线目录分隔符指定索引路径属性

在异类环境中，管理器节点的操作系统可以使用与对等节点的操作系统不同的约定来指定目录路径。由于您是在管理器节点上编辑 indexes.conf 文件，却将其分发到对等节点，因此出现了问题。

例如，如果您有一个 Windows 管理器节点和一组 Linux 对等节点，编辑文件的 Windows 管理器节点上指定 homePath 的常规方法是使用 Windows 反斜线约定作为目录分隔符，而分发文件的 Linux 对等节点则要求使用正斜线。

要处理这种可能性，最好始终使用正斜线来在索引路径属性中指定目录路径，而不管管理器节点和对等节点使用何种操作系统。例如：

```
homePath = $SPLUNK_DB/$_index_name/db
```

Splunk Enterprise 始终接受正斜线作为目录分隔符。

2. 将新 indexes.conf 文件分布到各对等节点

编辑完 indexes.conf 之后，需要将此文件分布到群集的一组对等节点上。要了解如何在所有对等节点上分布配置文件（包括 indexes.conf），请阅读“更新通用对等节点配置和应用”。

有关其他类型对等节点配置（包括应用分发）的信息，请阅读“对等节点配置概述”。

查看索引

要查看对等节点上的索引集，请单击管理器节点仪表板上的“索引”选项卡。请参阅“查看管理器节点仪表板”。

注意：新索引只在它包含一些数据之后才会出现于选项卡下方。换句话说，如果您在对等节点上配置了一个新索引，该索引的一行只在您向该索引发送数据后才会显示。

更新通用对等节点配置和应用

本主题中描述的对等节点更新进程可确保所有对等节点共享一组通用的关键配置文件。您必须手动调用该过程将通用文件（包括应用）分发和更新到对等节点。当对等节点加入群集时，该过程也会自动运行。

有关对等节点配置文件的信息，请参阅“在所有对等节点中管理通用配置”。该主题确切而又详细地介绍了哪些文件必须在所有对等节点之间保持一致。简单地说，在大多数情况下必须保持一致的配置文件包括 indexes.conf、props.conf 和 transforms.conf。其他配置文件可能也会保持一致，取决于您的系统要求。由于应用通常包括这些关键文件的多个版本，您还应在所有对等节点之间维护一组通用应用。

所有对等节点通用的配置文件和应用的集合，从管理器节点进行管理并通过单一操作分发到对等节点，该集合称为**配置软件包**。用于分发配置软件包的过程称为配置软件包方法。

要在所有对等节点之间分发新的或已编辑的配置文件或应用，您可以添加文件到管理器节点上的配置软件包，并指示管理器节点将文件分发到对等节点。

配置软件包的结构

配置软件包包括一组对所有对等节点通用的文件和应用。

在管理器节点上

在管理器节点上，配置软件包驻留于 $SPLUNK_HOME/etc/master-apps 目录下。该目录下的一组文件组成配置软件包。这些文件始终作为一个组分布到所有对等节点。此目录的结构如下所示：

```
$SPLUNK_HOME/etc/master-apps/
     _cluster/
          default/
          local/
     <app-name>/
     <app-name>/
     ...
```

请注意以下事项：

- 目录 /_cluster 是一个特殊位置，用于存放需要在所有对等节点之间分发的配置文件：
  - /_cluster/default 子目录包含 indexes.conf 的默认版本。请不要向此目录中添加任何文件，也不要更改其中的任何文件。此对等节点特定的默认 /_cluster/default 具有比标准默认 indexes.conf（位于 $SPLUNK_HOME/etc/system/default 下）更高的优先级。
  - 子目录 /_cluster/local 是存放要分发到对等节点的新的或已编辑的配置文件的位置。
- /<app-name> 子目录为可选。它们提供将任何应用分发到对等节点的方法。根据需要进行创建和填充。例如，要将 "appBestEver" 分发到对等节点，将该应用的副本放入其自己的子目录中：$SPLUNK_HOME/etc/master-apps/appBestEver。
- 要删除您之前分发给对等节点的应用，请从配置软件包中删除其目录。当您下次推送软件包的时候，该应用会从所有对等节点中删除。
- 管理器节点仅推送 master-apps 下子目录的内容。在 master-apps 下的任何独立文件将不直接推送。例如，它不推送独立

文件 /master-apps/file1。因此，请确保将任何独立配置文件放在 /_cluster/local 子目录中。

明确指示管理器节点要在何时将最新配置软件包分发到对等节点。另外，当某一对等节点在管理器节点中注册时（例如，当该对等节点加入群集时），管理器节点会将当前配置软件包分发到该对等节点。

管理器节点将软件包分发到对等节点时，它将分发整个软件包，覆盖以前分发到对等节点的任何配置软件包的全部内容。

master-apps 位置仅用于放对等节点文件。管理器节点不会使用该目录中的文件来解决其自己的配置需求。

在对等节点上

在对等节点上，分布后的配置软件包位于 $SPLUNK_HOME/etc/slave-apps 下。对等节点最初从管理器节点获取最新的软件包时，启用该对等节点后会立即创建此目录。除了顶层目录的名称不同以外，配置软件包的结构和内容与管理器节点上的配置软件包相同：

```
$SPLUNK_HOME/etc/slave-apps/
     _cluster/
          default/
          local/
     <app-name>/
     <app-name>/
     ...
```

将下载的文件保留在此位置中，不要编辑它们。如果稍后分发配置文件或应用的已更新版本到节点，它将覆盖 $SPLUNK_HOME/etc/slave-apps 中的所有早期版本。您会希望发生这种情况，因为群集中所有对等节点必须使用该目录中相同版本的文件。

出于相同原因，不要将任何文件或子目录直接添加到 $SPLUNK_HOME/etc/slave-apps。管理器节点每次重新分发配置软件包时都会覆盖此目录。

Splunk 软件评估配置文件时，$SPLUNK_HOME/etc/slave-apps/[_cluster|<app-name>]/local 子目录中的文件优先级最高。有关配置文件优先级的信息，请参阅*《管理员手册》*中的“配置文件优先级”。

不能通过配置软件包分发的设置

对等节点上的 $SPLUNK_HOME/etc/slave-apps 目录为只读。这是必要和有益的行为，因为每次分发新软件包的时候，目录会整个被覆盖。因此您会丢失对于目录设置所做的任何更改。群集也依赖于在所有对等节点中都相同的目录设置。

因此，如果您通过配置软件包的方法来分发配置的话，对等节点需要以某种方式自动更新，可以通过在 $SPLUNK_HOME/etc/apps 下创建新的应用版本来实现。由于在同一时间不能存在两个应用，这会在 splunkd.log 中产生“意外的重复应用”错误。

这一行为的常见原因是通过配置软件包分发 SSL 密码。重新启动后 Splunk Enterprise 用加密形式覆盖密码。但是如果您通过配置软件包的方法来分发配置的话，对等节点不能覆盖在 $SPLUNK_HOME/etc/slave-apps 下的软件包位置上的未加密密码。因此，在推送软件包后重新启动后，他们向 $SPLUNK_HOME/etc/apps 写入加密密码，其应用目录名与显示在 $SPLUNK_HOME/etc/slave-apps 下的名称一致。

例如，不推送 inputs.conf 中的下列设置：

```
[SSL]
password = <your_password>
```

如果是配置软件包中应用目录下称为“新应用”的设置，重新启动后对等节点会在 $SPLUNK_HOME/etc/apps 下产生一个“新应用”目录，并将设置放置此处。这样就能复制“新应用”的应用。

分发配置软件包中的 app.conf 的最佳做法

如果您添加新的 reload.<conf_file_name> = simple 参数到配置软件包中的 app.conf，在您将更新推送到新参数中引用的配置文件之前，您必须先将 app.conf 推送给对等节点。推送 app.conf 文件之后，随后对应用上下文中引用的配置文件进行更改将不再需要重新启动节点。

例如，如果您将 reload.inputs = simple 添加到 app.conf，然后将 app.conf 推送到对等节点，当您之后在特定应用上下文中将更新推送到 inputs.conf 时，对等节点无需重新启动即可重新加载。

要分发配置软件包中的 app.conf：

1. 针对您想要分发的每个应用，确定 app.conf 是否包含新的 reload.<conf_file_name> = simple 参数。
2. 从配置软件包中删除 reload.<conf_file_name> = simple 引用的任何配置文件。

3. 将包含 app.conf 的修改的配置软件包推送到对等节点集。请参阅“将软件包应用到对等节点”。
4. 将引用的配置文件返回至配置软件包中适当的应用目录。
5. 对您想要分发至对等节点的配置文件进行更改。
6. 将配置软件包推送到对等节点。

有关 reload.<conf_file_name> = simple 的更多信息，请参阅*管理员手册*中的 app.conf。

分布配置软件包

要在所有对等节点之间分发新的或已更改的文件和应用，请执行以下操作：

1. 准备文件和应用，并对它们进行测试。
2. 将文件和应用移入管理器节点上的配置软件包中。
3. （可选）验证软件包并检查重新启动。
4. 将软件包应用到对等节点。
   管理器节点将整个软件包推送到对等节点。这将覆盖对等节点的当前软件包的内容。

1. 为配置软件包准备文件和应用

对想要分发到对等节点的文件进行必要的编辑。建议在将文件和应用分发到一组对等节点之前，先在独立测试索引器上测试文件和任何应用，以确认这些文件和应用可正常使用。尽量将所有更新合并在单个软件包中，这样可降低对对等节点工作的影响。

有关如何配置文件的更多信息，请参阅“在所有对等节点中管理通用配置”和“在索引器群集中配置对等节点索引”。

如果配置软件包子目录包含定义新索引的 indexes.conf 文件，则必须将每个索引的 repFactor 属性显式设置为 auto。indexes.conf 文件（驻留在应用子目录中）以及 indexes.conf 文件（位于 _cluster 子目录中）都需要执行此操作。

有关更多信息，请参阅“indexes.conf repFactor 属性”。

2. 将文件和应用移入管理器节点上的配置软件包中

您准备好分发文件和应用时，将它们复制到管理器节点上的 $SPLUNK_HOME/etc/master-apps/：

- 将应用直接放在 master-apps 目录下。例如，$SPLUNK_HOME/etc/master-apps/<app-name>。
- 将独立文件放在 $SPLUNK_HOME/etc/master-apps/_cluster/local 子目录。

3. （可选）验证软件包并检查重新启动

在未应用软件包的情况下，您可以验证软件包并检查应用软件包是否需要重新启动对等节点。一旦确认软件包在所有对等节点中有效，您即可以在另一个步骤中应用该软件包。

验证对确保软件包顺利应用到所有对等节点很有用。验证过程中还提供对调试无效的软件包而言有用的信息。

对想要将软件包分发延迟到低活动时间或维持窗口，并想要避免可能发生的重新启动相关的索引或搜索干扰的管理员来说，检查重新启动功能很有用。请参阅“配置软件包更改后重新启动或重新加载？”

您可以使用 Splunk Web 或 CLI 验证软件包并检查重新启动。

使用 Splunk Web 验证软件包并检查重新启动

使用验证并检查重新启动按钮验证软件包并检查是否需要重新启动对等节点。此按钮功能与 CLI 命令功能相同：splunk validate cluster-bundle --check-restart。请参阅“使用 CLI 验证软件包并检查重新启动”。

要验证软件包并检查重新启动，请执行以下操作：

1. 在 Splunk Web 的管理器节点上，单击**设置 > 索引器群集化**。
   “管理器节点”仪表板打开。
2. 单击**编辑 > 配置软件包操作**。
3. 单击**验证并检查重新启动 > 验证并检查重新启动**。
   显示一条表示软件包验证和检查重启成功或失败的消息。

如果软件包验证和检查重新启动成功，则可将该软件包分发到对等节点。有关验证的软件包的信息显示在 UI 中，包括是否需要重新启动对等节点。UI 还会显示软件包校验和，您可用软件包校验和来识别和追踪活动的软件包，之前的软件包和最近检查重新启动的软件包。

您可以使用 Splunk Web 或 CLI 将配置软件包从管理器节点分发到对等节点。请参见 4. 应用软件包。

如果验证和检查重新启动失败，则不可将该软件包分发到对等节点。在此情况下，查看软件包详细信息可能有助于您解决问题。请确保配置软件包结构适合分发到对等节点。请参阅“配置软件包的结构”。

使用 CLI 验证软件包并检查重新启动

如只需验证软件包，请运行 `splunk validate cluster-bundle`：

```
splunk validate cluster-bundle
```

该命令将返回一条确认软件包验证已启动的消息。在某些验证失败的情况下，消息中还会说明失败的原因。

要验证软件包并检查是否需要重新启动，请包含 `--check-restart` 参数：

```
splunk validate cluster-bundle --check-restart
```

此版本的命令首先验证软件包。如果验证成功，则检查是否需要重新启动对等节点。

要查看软件包验证状态，请运行 `splunk show cluster-bundle-status` 命令：

```
splunk show cluster-bundle-status
```

此命令会提示您验证成功。如果验证失败，命令中会说明失败的原因。还会显示是否需要重新启动对等节点。

此示例为验证成功后 `splunk show cluster-bundle-status` 命令输出的内容：

```
master
        cluster_status=None
        active_bundle
               checksum=576F6BBB187EA6BC99CE0615B1DC151F
               timestamp=1495569737 (in localtime=Tue May 23 13:02:17 2017)
        latest_bundle
               checksum=576F6BBB187EA6BC99CE0615B1DC151F
               timestamp=1495569737 (in localtime=Tue May 23 13:02:17 2017)
        last_validated_bundle
               checksum=1E0C4F0A7363611774E1E65C8B3932CF
               last_validation_succeeded=1
               timestamp=1495574646 (in localtime=Tue May 23 14:24:06 2017)
        last_check_restart_bundle
                checksum=1E0C4F0A7363611774E1E65C8B3932CF
                last_check_restart_result=restart required
                timestamp=1495574646 (in localtime=Tue May 23 14:24:06 2017)
```

```
Peer 1          1D00A8C2-026B-4CAF-90D6-5D5D39445569          default
         active_bundle=576F6BBB187EA6BC99CE0615B1DC151F
         latest_bundle=576F6BBB187EA6BC99CE0615B1DC151F
         last_validated_bundle=1E0C4F0A7363611774E1E65C8B3932CF
         last_bundle_validation_status=success
         last_bundle_checked_for_restart=1E0C4F0A7363611774E1E65C8B3932CF
         last_check_restart_result=restart required
         restart_required_apply_bundle=0
         status=Up
...
```

`last_validated_bundle` 将识别新验证的软件包。请注意，此字段与 `active_bundle` 不同，后者将识别最近刚应用到节点上而且目前在各对等节点中启用的软件包。

`last_validation_succeeded=1` 字段会提示验证已成功。

在管理器节点上，`last_check_restart_result=restart required` 字段表示至少需要重新启动一个群集对等节点。

在对等节点上，`last_check_restart_result=restart required` 字段表示需要重新启动个别对等节点。

如果您验证了软件包却没有应用它，则管理器节点上的 `$SPLUNK_HOME/etc/master-apps` 目录的内容会不同于对等节点上的 `$SPLUNK_HOME/etc/slave-apps` 目录的内容，直到您应用了软件包。这对群集的操作没有影响，但能意识到存在差异很重要。

4. 将软件包应用到对等节点

要应用配置软件包到对等节点，您可以使用 Splunk Web 或 CLI。

如果软件包推送当前正在进行中，那么您无法启动配置软件包推送。

使用 Splunk Web 应用软件包

要将配置软件包应用到对等节点，请执行以下操作：

1. 在 Splunk Web 的管理器节点上，单击**设置 > 索引器群集化**。
   显示“管理器节点”仪表板。
2. 单击**编辑 > 配置软件包操作**。
   配置软件包操作仪表板打开，显示最近成功的软件包推送信息。

3. 单击**推送**。
   在某些情况下，弹出窗口将警告您分发可能启动所有对等节点的重新启动。有关哪些配置更改会导致对等节点重新启动的信息，请参阅“配置软件包更改后重新启动或重新加载？”。
4. 单击**推送更改**。
   屏幕将提供有关分发进度的信息。一旦分发完成或终止，屏幕将显示结果。
   - 如果分发成功，在每个对等节点成功验证软件包后，管理器节点将在必要时协调所有对等节点的滚动重新启动。
   - 如果终止分发，它将显示哪些对等节点无法接收分发。每个对等节点必须成功收到并应用分发。任何不成功的对等节点都无法应用软件包。

   推送成功后，对等节点将使用一组新的配置，这些配置现位于其本地 `$SPLUNK_HOME/etc/slave-apps` 中。

   将这些文件保留在 `$SPLUNK_HOME/etc/slave-apps` 中。

关于分发过程的详细信息，请参阅关于通过 CLI 应用软件包的下一部分。

使用 CLI 应用软件包

1. 要将配置软件包应用到对等节点，请在管理器节点上运行以下 CLI 命令：

   ```
   splunk apply cluster-bundle
   ```

   主节点会发出此警告消息进行响应：

   ```
   Caution: Under some circumstances, this command will initiate a rolling restart
   of all peers. This depends on the contents of the configuration bundle. For
   details, refer to the documentation. Do you wish to continue? [y/n]:
   ```

   有关哪些配置更改会导致滚动重新启动的信息，请参阅“配置软件包更改后重新启动或重新加载？”。

2. 要继续进行，需要以 `y` 响应此消息。通过将 `--answer-yes` 标记附加到该命令，您可以避免收到此消息：

   ```
   splunk apply cluster-bundle --answer-yes
   ```

   `splunk apply cluster-bundle` 命令会导致管理器节点将新的配置软件包分布到各个对等节点，这些对等节点随后会个别验证软件包。在此过程中，每个对等节点验证软件包中所有 `indexes.conf` 文件的设置。在所有对等节点成功验证软件包后，管理器节点将在必要时协调所有对等节点的滚动重新启动。

   下载和验证过程通常只需要几秒钟即可完成。如果任何对等节点无法验证配置软件包，它会向管理器节点发送消息，管理器节点会在 Splunk Web 的相应仪表板中显示这一错误。该过程不会继续到下一个阶段，即重新加载或重新启动对等节点，除非所有对等节点均已成功验证软件包。

   如果验证没有成功，则必须修复管理器节点提示的问题并重新运行 `splunk apply cluster-bundle`。

   一旦验证完成，管理器节点将告诉对等节点重新加载，或在必要时启动所有对等节点的滚动重新启动。有关滚动重新启动如何工作的详细信息，请参阅“执行索引器群集的滚动重启”。要为软件包推送触发的滚动重新启动将可搜索滚动重新启动设为默认模式，请参阅“结合使用可搜索的滚动重新启动与配置软件包推送”。

   过程完成后，对等节点将使用一组新的配置，这些配置位于其本地 `$SPLUNK_HOME/etc/slave-apps` 中。

   将这些文件保留在 `$SPLUNK_HOME/etc/slave-apps` 中。

一旦分发应用到一组对等节点，您将使用 Splunk Web 以常用方式启动和管理每个对等节点。请参阅*《管理员手册》*中的“管理应用配置和属性”。

`apply cluster-bundle` 命令将使用可选标记 `--skip-validation`，以在验证流程出现问题的情况下使用。您应仅在 Splunk 支持的指示并确信软件包有效后使用本标记。不要使用本标记以包围验证流程，除非您知道正在做什么。

您可以在不应用软件包的情况下对其进行验证。如果需要调试一些验证问题，此操作很有用。请参见 3.（可选）验证软件包。

使用 CLI 查看软件包推送的状态

要查看群集软件包推送的进行情况，请从管理器节点运行以下命令：

```
splunk show cluster-bundle-status
```

此命令告诉您软件包验证是成功还是失败。它还指示每个对等节点的重新启动状态。

配置软件包更改后重新启动或重新加载

配置软件包中一些文件的更改要求对等节点重新启动。在其他情况下，对等节点会重新加载配置文件，避免创建索引或搜索过程中出现中断情况。在对等节点上的软件包重新加载阶段确定是否需要重启，并仅在必要时指示管理器节点启动对等节点的滚动重新启动。

确定何时重新启动或重新加载

以下总结了将配置更改推送到索引器群集对等节点时的重新加载和重新启动行为：

重新加载发生在：

- 您对 `app.conf` 的 `[triggers]` 段落中列出的任何配置文件进行了更改，但对 `indexes.conf` 进行某些更改除外。
- 在 `indexes.conf` 中做了这些更改：
  - 添加新的索引段落

    - 启用或禁用不带数据的索引
    - 更改“确定 indexes.conf 的哪种更改需要重新启动”中未列出的需要重新启动的属性。
- server.conf 文件在 shclustering 段落中仅包含 conf_replication_include <conf_file_name> = true 属性，并且该文件不包含其他属性。

重新启动发生在：

- 您对 app.conf 的 [triggers] 段落中未列出的任何配置文件进行了更改。
- 执行了“确定 indexes.conf 的哪种更改需要重新启动”中描述的任何一个对 indexes.conf 的更改。
- 对包含除 conf_replication_include <conf_file_name> = true 属性之外的其他属性的 server.conf 文件进行了任何更改。
- 从配置软件包中删除了现有应用。

更多信息，请参阅*管理员手册*中的“更改配置文件之后何时重新启动 Splunk Enterprise”。

app.conf 中的配置文件重新加载触发条件

Splunk 应用程序可以包含 Splunk Enterprise 核心配置文件和自定义配置文件的组合，例如由应用程序开发人员为 Splunkbase 上的私有应用程序和公共应用程序创建的文件。当您将配置更改推送到群集对等节点时，这些配置文件是否会重新加载取决于 app.conf 中的重新加载触发条件设置。

Splunk Enterprise 核心配置文件默认情况下会重新加载，但上节中概述的例外情况除外。这些文件在 $SPLUNK_HOME/etc/system/default/app.conf 的 [triggers] 段落中指定了预定义的重新加载触发条件，这些条件让它们可以自动重新加载。

自定义配置文件定义为，在 $SPLUNK_HOME/etc/system/README 中没有对应的 .spec 文件的任何配置文件。这包括在第三方应用程序中的自定义配置文件，例如 aws_settings.conf、service_now.conf、eventgen.conf 等。

默认情况下，所有自定义配置文件都会重新加载，除非该文件在 app.conf 中已经定义了自定义重新加载触发条件。例如，Splunk Security Essentials 应用程序，app.conf 包含以下自定义重新加载触发条件：reload.ssenav = http_get /SSEResetLocalNav. 当您为某个自定义配置文件推送配置更改，而该配件文件在 app.conf 中已经定义了自定义重新加载触发条件时，Spunk 软件会尝试采用自定义重新加载触发条件设置。如果该自定义重新加载触发条件失败，则将发生滚动重启。

如果自定义配置文件没有在 app.conf 中指定重新加载触发条件，则默认行为是为未知配置重新启动。如果不需要重新启动，您可以将 app.conf 中的 conf 级别触发条件设为 reload.<conf_file_name> = simple。

有关自定义配置文件的重新加载触发条件设置的详细信息，请参阅*《管理员手册》*中的 app.conf。

有关最常用的可重新加载的应用程序和 .conf 文件的列表，请参阅 Splunk Cloud 文档中的“常见应用程序和 .conf 文件的重启与重新加载行为”。

input.conf 的段落级重新加载触发条件

app.conf 中的段落级重新加载触发条件只会让重新加载特定的配置文件段落，这些段落在您执行配置软件包推送时发生了更改。这让您可以根据配置文件中会发生更改的具体段落，进行更有效的配置更新。

段落级重新加载当前仅适用于 inputs.conf 中的段落子集。当指定段落发生更改后，inputs.conf 中的特定段落（在 app.conf 的 [triggers] 段落下具有 reload.<conf_file_name>.<conf_stanza_prefix> 条目）会重新加载。

对段落级重新加载条目中未指定的任意 inputs.conf 段落进行的更改都会触发滚动重启。

仅当将更改推送到索引器群集上下文中的配置软件包时，inputs.conf 的段落级重新加载才适用。

以下段落可在 inputs.conf 中重新加载：

.conf 文件名	段落前缀	重新加载或重启
inputs.conf	http	reload
inputs.conf	script	reload
inputs.conf	monitor	reload
inputs.conf	<modular_input>	reload
inputs.conf	batch	reload

有关段落级重新加载触发条件的详细信息，请参阅*《管理员手册》*中的 app.conf。

禁用 app.conf 中的重新加载触发条件

您可以为 app.conf 中的任意重新加载触发条件条目指定值 never，通过此方式禁用 .conf 级重新加载触发条件和段落级重新加载触发条件。发生配置更改时，任何值为 never 的重新加载触发条件条目都将触发滚动重启。如果出于任何原因您希望特定配置更改可以触发滚动重启，这点就很有用。

有关配置重新加载触发条件的更多信息，请参阅*《管理员手册》*中的 app.conf。

结合使用可搜索的滚动重新启动与配置软件包推送

可搜索选项允许您执行对等节点滚动重新启动，尽量减少正在进行的搜索中断情况。您可为配置软件包推送触发的所有滚动重新启动，将 server.conf 中的可搜索的滚动重新启动设为默认模式。请参阅“为软件包推送将可搜索滚动重新启动设为默认模式”查看说明。

有关更多信息，请参阅“执行索引器群集的滚动重启”。

回滚配置软件包

您可以使配置软件包回滚为之前的版本。此操作可以让您从一个错误配置的软件包中恢复。

回滚操作将对等节点上最近刚应用的配置软件包切换为之前应用的软件包。不能回滚到除之前软件包以外的软件包。

例如，假设对等节点有一个启用的配置软件包 "A"，而您应用了一个配置软件包 "B"，则它变为新的启用软件包。如果您发现 B 有问题，则可以回滚为软件包 A，然后对等节点会使用 A 作为它们的启用软件包。如果进行第二次回滚，对等节点将再次返回到软件包 B。如果进行第三次回滚，软件包将再次返回到 A，依此类推。回滚操作会一直切换最近的两个软件包。

您可以使用 Splunk Web 或 CLI 回滚配置软件包。

使用 Splunk Web 回滚配置软件包

1. 在 Splunk Web 的管理器节点上，单击**设置 > 索引器群集化**。
   “管理器节点”仪表板打开。
2. 单击**编辑 > 配置软件包操作**。
3. 单击**回滚**。
   显示表示回滚成功或失败的消息。

如果之前不存在配置软件包，则禁用“回滚”按钮。

使用 CLI 回滚配置软件包

要回滚配置软件包，请在管理器节点上运行此命令：

```
splunk rollback cluster-bundle
```

如同 splunk apply cluster-bundle，此命令在必要时启动对等节点的滚动重新启动。

您可以使用 splunk show cluster-bundle-status 命令来确定当前启用的软件包。您可以使用 cluster/manager/info 端点来获取当前启用和之前启用的软件包的信息。

如果 master-apps 文件夹被破坏导致回滚失败，一条指定失败和解决方法的消息会出现在管理器节点的仪表板上以及 splunkd.log 中。要修复，请遵循消息中的指示。操作包括删除指示失败的 $SPLUNK_HOME/etc/master-apps.dirty marker 文件，并按照消息中指定的方式在启用的软件包上进行手动复制。

在 Windows 上，如果有打开的文件正在处理 $SPLUNK_HOME/etc/master-apps 及其内容，则回滚操作会失败。

对等节点启动时软件包的分布

最初将 Splunk 实例配置为对等节点后，您必须手动重启该实例才能加入群集。详参“启用对等节点”中的介绍。在重新启动过程中，对等节点与管理器节点建立连接，下载当前配置软件包，本地验证软件包，然后再次重新启动。只有软件包验证成功时，对等节点才会加入群集。脱机对等节点恢复联机时也会出现相同过程。

如果验证失败，则用户必须修复错误并从管理器节点运行 splunk apply cluster-bundle。

使用部署服务器将应用分布到管理器节点

尽管无法使用部署服务器直接分发应用到对等节点，但是您可将它用于分发应用到管理器节点的配置软件包位置。一旦应用在位置中，管理器节点可通过本主题介绍的配置软件包方法分发它们到对等节点。

除了部署服务器，您也可以使用第三方分布式配置管理软件，例如 Puppet 或 Chef，来分发应用给管理器节点。

要使用部署服务器分发文件到管理器节点上的配置软件包：

1. 将管理器节点配置成部署服务器的客户端，如*《更新 Splunk Enterprise 实例手册》*的“配置部署客户端”中所述。
2. 在管理器节点上，编辑 deploymentclient.conf 并设置 repositoryLocation 属性为 master-apps 位置：

   ```
   [deployment-client]
   serverRepositoryLocationPolicy = rejectAlways
   repositoryLocation = $SPLUNK_HOME/etc/master-apps
   ```
3. 在部署服务器上，创建并填入一个或多个部署应用，以便下载到管理器节点的配置软件包。确保应用遵照配置软件包的结构要求，如本主题先前所列出。有关创建部署应用的信息，请参阅*《更新 Splunk Enterprise 实例》*中的“创建部署应用”。
4. 创建一个或多个将管理器节点映射到部署应用的服务器类。有关创建服务器类的信息，请参阅*《更新 Splunk Enterprise 实例》*中的“创建服务器类”。
5. 每个服务器类必须包括 stateOnClient = noop 设置：

   ```
   [serverClass:<serverClassName>]
   stateOnClient = noop
   ```

   切勿在应用段落级别覆盖该设置。
6. 将应用下载到管理器节点上。一旦管理器节点收到配置软件包中的新的或更新的部署应用，您可以使用当前主题介绍的方法分发软件包到对等节点。

   采取步骤确保管理器节点在收到部署应用后不会自动重新启动。尤其是，当定义部署应用行为时，不要更改 restartSplunkd 设置的默认值 "false"（位于 serverclass.conf 中）。如果正在使用转发器管理定义您的服务器类，则确保没有勾选“编辑应用”屏幕上的“重新启动 splunkd”字段。

有关部署服务器和如何执行必要的各种操作的详细信息，请阅读*《更新 Splunk Enterprise 实例》*手册。

按对等节点管理配置

所有对等节点的大多数配置必须相同。请参阅“在所有对等节点中管理通用配置”。对于有限的应用，比如说测试，您可按节点处理一些配置。

配置数据导入

建议使用转发器来处理对等节点的数据导入。有关配置此过程的信息，请参阅“使用转发器在索引器群集中获取数据”。

如果您要将数据直接输入到某一对等节点，而不使用转发器，可以在此对等节点上配置您的输入，方式与索引器的输入配置方式相同。有关更多信息，请参阅*《数据导入》*手册中的“配置输入”。

重要提示：尽管您可以按对等节点配置输入，但需考虑具体需求是否允许在所有对等节点中使用一组输入。如果所有数据都通过转发器传送，并且所有对等节点上的接收端口都相同，那么这就应该允许使用一组输入。如果是这样，您可以使用管理器节点管理通用 inputs.conf 文件，如“更新通用对等节点配置和应用”所述。

将索引添加到单个对等节点

如果您需要将索引添加到某单个对等节点，可通过在该对等节点上创建单独的indexes.conf 文件实现此目的。但是，新索引中的数据将仅保留在该对等节点上，并且将不会被复制。主要相关用例是执行某种本地测试或监视，可能涉及将某个应用仅下载到该单个对等节点。特定于对等节点的 indexes.conf 可以补充（但不能取代）所有对等节点获得的通用文件版本。

如果您为某单个对等节点创建 indexes.conf 的一个版本，可以将该版本放入索引器的任何可接受位置中，请参阅*《管理员手册》*中“关于配置文件”和“配置文件目录”的阐述。此文件不能放在 $SPLUNK_HOME/etc/slave-apps 之下，因为这是配置软件包在对等节点上所驻留的位置。如果将此文件置于此处，那么当该对等节点下次下载配置软件包时，此文件将会被覆盖。

重要提示：如果添加本地索引，请将其 repFactor 属性设置为默认值 0。不要设置为 auto。如果将其设置为 auto，该对等节点会试图将索引的数据复制到群集中的其他对等节点。由于不会为新索引配置其他对等节点，所以在其他对等节点上没有存储复制的数据的位置，从而导致各种问题，有的可能非常严重。此外，当管理器节点下次尝试向对等节点推送配置软件包时，未正确配置索引的对等节点将返回软件包验证错误给管理器节点，阻止管理器节点将软件包成功应用到这组对等节点。

进行其他配置更改

如果您需要特定于单个对等节点进行某些其他配置更改，可以使用适用于任何 Splunk Enterprise 例（群集或非群集）的正常方式配置这些对等节点。可以使用 Splunk Web 或 CLI，也可以直接编辑配置文件。

重新启动对等节点

与任何索引器一样，在更改某一对等节点的配置之后有时需要重新启动该对等节点。但与非群集索引器不同，不要使用 CLI 命令 splunk restart 重新启动该对等节点。而是应使用 Splunk Web 中的重新启动功能。有关如何重新启动群集对等节点的详细信息，请阅读“重新启动单个对等节点”。

有关需要重新启动的配置更改的信息，请参阅“修改 server.conf 后重新启动”和“配置软件包更改后重新启动或重新加载”。

配置搜索头

搜索头配置概述

在索引器群集中的搜索头配置可分为这几类：

- 群集节点配置。在索引器群集的初始部署期间进行搜索头节点的基本配置。您可之后再编辑配置。
- 高级功能和拓扑。这些功能，比如说安装软件包，对所有搜索头可用，无论它们是否参与索引器群集的活动。
- 合并搜索。您可在多站点群集之间或群集和非群集搜索节点之间合并搜索。

重要提示：本章介绍了在索引器群集中充当节点的独立搜索头。有关如何将作为**搜索头群集**成员的搜索头整合进索引器群集的信息，请参阅*《分布式搜索》*手册中的“通过索引器群集集成搜索头群集”。此外，请参阅*《分布式搜索》*手册中的“配置搜索头群集化”一章。

群集节点配置

当对索引器群集进行初始部署时，会将 Splunk Enterprise 实例作为索引器群集的搜索头进行基本的配置。您可之后再编辑配置。

执行初始配置

您可以在启用其他群集节点的同时配置和启用搜索头，如“启用搜索头”所述。群集的对等节点集将成为搜索头的**搜索节点**。对于基本功能，您无需设置任何其他配置。

编辑配置

为特定的群集编辑基本的搜索头配置有两个主要原因：

- **重定向搜索头到同一群集的另一个管理器节点。**当管理器节点发生故障，但是您的该群集拥有可以重定向搜索头到的备用管理器节点，则这非常有用。关于备用管理器节点的信息，请参阅“在索引器群集中替换管理器节点”。
- **更改该群集的搜索头的安全密钥。**如果您还为群集中的所有其他节点更改，则仅更改安全密钥。群集中的所有实例的密钥必须相同。

要编辑搜索头的群集节点配置，请使用下面方法中的一种：

- 从 Splunk Web 中的搜索头节点仪表板上编辑配置。请参阅“使用仪表板配置搜索头”。
- 编辑搜索头的 server.conf 文件。请参阅“使用 server.conf 配置搜索头”。
- 使用 CLI。请参阅“使用 CLI 配置搜索头”。

配置多站点搜索头

关于配置多站点搜索头时的附加项和不同点，请参阅“在多站点索引器群集中实现搜索相关性”和“使用 server.conf 配置多站点索引器群集”。

高级功能和拓扑

要实现分布式搜索某些更高级的功能，例如安装软件包，必须在该搜索头上编辑 distsearch.conf。

关于如何执行高级配置的说明，请阅读*《分布式搜索》*手册。该手册侧重于介绍非群集索引器环境，但是除了此处介绍的内容，您可以使用与索引器群集同样的方式配置大多数高级搜索头功能。

运行于索引器群集的搜索头与运行于非群集索引器的搜索头的比较

运行于索引器群集的搜索头与那些运行于非群集索引器的搜索头的大多数设置和功能是相同的。

主要差异在于，对于索引器群集而言，搜索头和搜索节点会在群集启用过程中相互间自动连接。您无需在 distsearch.conf 中执行任何配置，即可启用自动发现。

distsearch.conf 中有几个属性对于索引器群集中的搜索头无效。索引器群集中的搜索头忽略以下属性：

```
servers
disabled_servers
heartbeatMcastAddr
heartbeatPort
heartbeatFrequency
ttl
checkTimedOutServersFrequency
autoAddServers
```

与在非群集索引器中运行一样，搜索头对搜索节点的访问也是通过公共密钥验证进行控制。但是，不需要手动分布密钥。索引器群集中的搜索头会自动将其公共密钥推送到搜索对等节点。

已安装软件包和搜索节点配置

大多数 `distsearch.conf` 设置仅对于搜索头有效。但是，要实施已安装的软件包，还需要将一个小型的 `distsearch.conf` 文件分布到搜索节点。对于索引器群集，应该使用管理器节点将此文件分布到各对等节点。有关如何使用管理器节点管理对等节点配置的信息，请阅读本手册中的“更新通用对等节点配置和应用”。有关如何配置已安装软件包的信息，请阅读*《分布式搜索》*手册中的“已安装知识软件包复制”。

分布式搜索页面和索引器群集一起工作的方式

切勿使用 Splunk Web 上的“分布式搜索”页面来配置索引器群集中的搜索头或添加对等节点到群集中。然而，您可以使用该页面查看搜索对等节点的列表。

合并搜索

要跨多个索引器群集搜索，请参阅“跨多个索引器群集搜索”。

要跨群集化和非群集化搜索对等节点进行搜索，请参阅“跨群集和非群集搜索对等节点搜索”。

使用仪表板配置搜索头

您可通过仪表板编辑现有搜索头节点的配置。访问仪表板：

1. 单击 Splunk Web 右上角的**设置**。

2. 在**分布式环境**组中，单击**索引器群集化**。

仪表板包括大量菜单项和影响配置的操作。

有关首次使用此仪表板启用搜索头的信息，请参阅“启用搜索头”。

关于使用仪表板查看索引器群集状态的信息，请参阅“查看搜索头仪表板”。

为特定的群集更改配置

要更改某个特定索引器群集的搜索头的配置，选择群集行上的**编辑配置**操作。

- 要更改管理器节点，则编辑**管理器 URI** 字段。
- 要更改安全密钥，则编辑**安全密钥**字段。

加入另一个索引器群集

要将搜索头连接到另一个群集，请参阅“跨多个索引器群集搜索”。

从群集删除搜索头

要从索引器群集删除搜索头，选择该群集行上的**删除群集**操作。这将取消搜索头与该群集的关联，但是保留其与所有其他群集的连接（如果有的话）。

要从所有群集删除搜索头，从仪表板右上角的**编辑**菜单选择**禁用群集化**。

其他编辑

如果需要更改本实例为一些其他节点类型（如对等节点类型），您可以通过仪表板右上角的**编辑**按钮完成。

您不太可能会希望更改已启用群集中的节点的节点类型。进行更改之前，仔细考虑后果。

编辑按钮对多站点群集来说是禁用的。

使用 server.conf 配置搜索头

前提条件

在阅读本主题之前，请参阅“使用 server.conf 配置索引器群集”。它介绍了对于所有群集节点类型都普遍存在的配置问题。

启用搜索头

下面的示例显示了一些基本设置，您必须在启用搜索头节点时就配置它们。显示的配置属性与 Splunk Web 中的**启用群集化**页面上的字段相对应。

```
[clustering]
manager_uri = https://10.152.31.202:8089
mode = searchhead
pass4SymmKey = whatever
```

该示例指定了以下内容：

- 此搜索头的群集管理器节点位于 10.152.31.202:8089。
- 实例为群集搜索头。
- 安全密钥为 "whatever"。

编辑搜索头设置

如有必要，您可以稍后更改这些设置。例如，要更改群集的安全密钥，可在每个节点上更改 pass4SymmKey 的值。

您也可配置搜索头，在多索引器群集之间或群集和非群集搜索对等节点之间进行搜索。有关更多信息，请参阅：

- 跨多个索引器群集搜索
- 跨群集和非群集搜索对等节点搜索。

使用 CLI 配置搜索头

首先阅读

在阅读本主题前，请参阅：

- “使用 CLI 配置和管理索引器群集”。本主题介绍使用 CLI 进行基本的索引器群集配置。它提供关于所有群集节点类型的普遍问题的详细信息。

启用搜索头

下面的示例显示了基本的设置，这些设置通常在启用搜索头时配置。配置属性与 Splunk Web 中的**启用群集化**页面上的字段相对应。

要启用一个实例作为搜索头，请将 mode 设置为 "searchhead"。您还需要设置 manager_uri 和用于整个群集的安全密钥 (secret)：

```
splunk edit cluster-config -mode searchhead -manager_uri https://10.160.31.200:8089 -secret your_key

splunk restart
```

-secret 标记编辑 server.conf 的 [clustering] 段落中的 pass4SymmKey 设置。

编辑搜索头设置

您还可以稍后使用 CLI 来编辑配置。

重要提示：当您初次启用一个搜索头时，请使用 splunk edit cluster-config 命令。而要更改搜索头配置，则必须改用 splunk edit cluster-master 命令。

例如，要更改安全密钥 (secret)，使用此命令：

```
splunk edit cluster-master https://10.160.31.200:8089  -secret newsecret123
```

重要提示： splunk edit cluster-master 命令总是把当前管理器节点 URI:port 值作为其初始值。例如，此命令通过为 -master_uri 参数设置一个新的值，将搜索头连接到一个不同的管理器节点，但是它为旧的管理器节点提供了一个值作为其初始参数：

```
splunk edit cluster-master https://10.160.31.200:8089  -master_uri https://10.160.31.55:8089
```

有关可配置设置的列表，请参阅 CLI 群集化帮助，以及 server.conf 规范文件。

跨多个索引器群集搜索

您可配置一个搜索头，在多索引器群集之间搜索。使用的方法取决于群集是单个站点还是多站点。

为单站点索引器群集配置多群集搜索

配置多群集搜索：

1. 以正常的方式为其中一个群集配置搜索头，如“启用搜索头”中所述。

2. 在新群集中，将搜索头指向管理器节点。您可用 Splunk Web、通过 CLI 或编辑搜索头的 server.conf 文件进行配置。

在 Splunk Web 中

在 Splunk Web 中，从搜索头仪表板中配置多群集搜索：

1. 选择仪表板右上角的**添加要搜索的群集**按钮。

2. 填写弹出窗口中的字段：

- **管理器 URI**。输入管理器节点 URI，包括它的管理端口。例如：https://10.152.31.202:8089。
- **安全密钥**。这是验证群集管理器节点、对等节点与搜索头之间通信的密钥。群集中所有节点的密钥必须相同。在此输入新群集的安全密钥。搜索头各群集的密钥可能会不同。

要从群集中删除搜索头，请参阅“从群集中删除搜索头”。

通过 CLI

在 CLI 中，您可以使用这些命令配置多群集搜索：

```
splunk add cluster-master <master_uri:port>
splunk edit cluster-master <master_uri:port>
splunk remove cluster-master <master_uri:port>
splunk list cluster-master
```

在运行这些命令后，不需要重新启动搜索头。

例如，要将搜索头添加到一个群集中，此群集的管理器节点位于 https://10.160.31.200:8089，则运行此命令：

```
splunk add cluster-master https://10.160.31.200:8089 -secret your_key
```

有关任何命令的更多信息，请参阅其 CLI 帮助。

通过编辑 server.conf

您可以在该搜索头的 server.conf 文件中配置多群集搜索。具体作法为，在 manager_uri 属性中指定以逗号分隔的管理器节点参考列表，后跟每个管理器节点的各个段落。例如：

```
[clustering]
mode = searchhead
manager_uri = clustermanager:east, clustermanager:west
```

```
[clustermanager:east]
manager_uri=https://SplunkManager01.example.com:8089
pass4SymmKey=someSecret

[clustermanager:west]
manager_uri=https://SplunkManager02.example.com:8089
pass4SymmKey=anotherSecret
```

在本示例中，搜索头与 SplunkManager01 通信时使用 pass4SymmKey "someSecret"，与 SplunkManager02 通信时则使用 pass4SymmKey "anotherSecret"。

编辑 server.conf 后，必须重新启动搜索头，才能使更改生效。

有关配置多群集搜索的详细信息，请参阅 server.conf 规范文件。

为多站点索引器群集配置多群集搜索

搜索头能在多个多站点群集之间或单个站点和多站点群集的组合之间搜索。要进行这样的配置，当连接到多站点群集时，需要指定搜索头的 site 属性。

通过 CLI

在 CLI 中，您可以使用 splunk add cluster-master 命令配置多群集搜索。当添加一个多站点群集时，包括搜索头的 site 值：

```
splunk add cluster-master <master_uri:port> -site site<n>
```

运行此命令后，您不需重新启动搜索头。

通过编辑 server.conf

要为一个多站点群集配置多群集搜索，需要设置两个多站点特定的属性：site 和 multisite。这些属性的位置有不同，取决于一些因素。

如果搜索头仅在多站点群集之间搜索，且在每个群集中，搜索头都在同一站点，则将 site 属性放在 [general] 段落下，将 multisite 属性放在每个 [clustermanager] 段落下：

```
[general]
site=site1

[clustering]
mode = searchhead
manager_uri = clustermanager:multieast, clustermanager:multiwest

[clustermanager:multieast]
multisite=true
manager_uri=https://SplunkManager01.example.com:8089
pass4SymmKey=someSecret

[clustermanager:multiwest]
multisite=true
manager_uri=https://SplunkManager02.example.com:8089
pass4SymmKey=anotherSecret
```

如果搜索头仅在多站点群集之间搜索，且在每个群集中，搜索头位于不同站点，则将 site 和 multisite 属性都放在 [clustermanager] 段落下：

```
[clustering]
mode = searchhead
manager_uri = clustermanager:multieast, clustermanager:multiwest

[clustermanager:multieast]
multisite=true
manager_uri=https://SplunkManager01.example.com:8089
pass4SymmKey=someSecret
site=site1
```

```
[clustermanager:multiwest]
multisite=true
manager_uri=https://SplunkManager02.example.com:8089
pass4SymmKey=anotherSecret
site=site2
```

如果搜索头在单个站点和多站点群集的组合之间搜索，则对于任何多站点群集，将 site 和 multisite 属性都放在 [clustermanager] 段落下。在此示例中，搜索头在两个群集之间搜索，只有一个是多站点群集：

```
[clustering]
mode = searchhead
manager_uri = clustermanager:multi, clustermanager:single

[clustermanager:multi]
multisite=true
manager_uri=https://SplunkManager01.example.com:8089
pass4SymmKey=someSecret
site=site1

[clustermanager:single]
manager_uri=https://SplunkManager02.example.com:8089
pass4SymmKey=anotherSecret
```

编辑 server.conf 后，必须重新启动搜索头，才能使更改生效。

关于多站点群集配置的信息，请参阅“使用 server.conf 配置多站点索引器群集”。

跨群集和非群集搜索对等节点搜索

您可以跨群集和非群集搜索对等节点搜索：

1. 以标准方式配置索引器群集搜索头，如“启用搜索头”所述。

2. 使用 Splunk Web 或 CLI 添加一个或多个非群集搜索对等节点，如*《分布式搜索》*中的“添加搜索对等节点到搜索头”所述。

请注意以下事项：

- 该过程假设您正启动一个新的 Splunk Enterprise 实例并想将其启用为一个索引器群集和一个或多个非群集索引器的搜索头。如果此实例已经是两种角色中其中一个的搜索头，则只需为该角色启用此实例。例如，如果此搜索头已经是某个索引器群集的一部分，则只需执行步骤 2。
- 您必须通过 Splunk Web 或 CLI 指定非群集搜索对等节点。出于验证问题，您无法通过直接编辑 distsearch.conf 指定搜索对等节点。当添加带 Splunk Web 或 CLI 的搜索节点时，搜索头将提示您导入公共密钥证书。当通过直接编辑 distsearch.conf 添加搜索节点时，没有其他办法获得这些证书。有关公共密钥和分布式搜索的更多信息，请参阅*《分布式搜索》*中的“添加搜索对等节点到搜索头”。

- 索引器可以是一个群集节点，在某种情况下，它自动成为其群集的搜索头的一个搜索节点，或者索引器也可以是一个非群集搜索节点，在搜索头的 distsearch.conf 文件中有一个条目。但是它不能两者都是。如果您错误地将索引器配置成一个群集节点和一个非群集搜索节点，则 Splunk Web 中的搜索头的分布式搜索页面将包含该节点的两个条目，其中一个条目的状态会是，“群集的节点成员和 distsearch.conf”。要修复的话，可在 distsearch.conf 中禁用或删除该节点的条目。

- 这种类型的搜索和搜索头合并不能兼容。

部署和配置多站点索引器群集

多站点索引器群集部署概述

在阅读本主题前，请参阅：

- “索引器群集部署概述”。本主题提供关于同时部署单个站点和**多站点**索引器群集的一般概述。您正在阅读的主题仅介绍多站点的差异。

重要提示：本章假定您正在多站点索引器群集中部署独立搜索头。有关如何整合作为**搜索头群集**成员的搜索头的信息，请参阅*《分布式搜索》*手册中的“通过索引器群集集成搜索头群集”。

从单个站点群集迁移

要从单个站点迁移到多站点索引器群集，请阅读“将索引器群集从单个站点迁移到多站点”。

部署多站点索引器群集

要部署一个多站点群集，则要为每个站点配置一组节点：

- 单个的管理器节点驻留在其中一个站点上，控制整个多站点群集。
- 一组对等节点驻留在每个站点上。
- 一个搜索头驻留在每个站点，用来搜索群集数据。如果想要所有搜索都在本地，则必须在每个站点安装一个搜索头。这被称为**搜索相关性**。

例如，要用三个对等节点和每个站点上的一个搜索头设置一个两站点群集，需安装和配置如下实例：

- 在某个站点上（站点 1 或站点 2）的一个管理器节点
- 站点 1 上的三个对等节点
- 站点 2 上的三个对等节点
- 站点 1 上的一个搜索头
- 站点 2 上的一个搜索头

注意：管理器节点本身实际上并不是任何站点的成员，除了其物理位置之外。但是，每个管理器节点有一个内置搜索头，此搜索头需要您在管理器节点的配置中设置一个站点属性。必须为管理器节点指定一个站点，即便永远不使用其内置搜索头。注意，搜索头仅作测试。不要将其用于生产。

配置多站点节点

要部署和配置多站点群集节点，则必须直接编辑 server.conf 或使用 CLI。不能使用 Splunk Web。

多站点特定的配置设置

当部署一个多站点群集时，和单个站点一样进行配置，同时还有一些其他设置去指定一组站点和站点之间复制的以及可搜索副本的位置。

在管理器节点上，您：

- 为多站点启用群集。
- 为群集枚举一组站点。
- 设置多站点复制因子。
- 设置多站点搜索因子。
- 必要时调整单站点复制因子和搜索因子。请参阅“多站点群集不满足其复制或搜索因子”。

在每个群集节点，应：

- 识别节点驻留的站点。

使用 server.conf 进行配置

关于使用 server.conf 配置多站点管理器节点的信息，请参阅“使用 server.conf 配置多站点索引器群集”。

使用 CLI 配置

关于使用 CLI 配置多站点管理器节点的信息，请参阅“使用 CLI 配置多站点索引器群集”。

结合使用索引器发现与多站点群集

如果您正使用索引器发现来连接转发器到对等节点，您必须为每个转发器分配一个站点。请参阅“在多站点群集中使用索引器发现”。

在多站点索引器群集中实现搜索相关性

多站点索引器群集化的重要好处之一是，它允许您配置一个群集，这样搜索头可以仅从存储在本地的站点获得搜索结果。这样减少了网络流量，同时仍然提供了对整个数据组的访问，因为每个站点包含数据的全部副本。这一好处被称为**搜索相关性**。

例如，您在 California 有两个数据中心，一个在 San Francisco，另一个在 Los Angeles。您设置一个两站点群集，每个站点对应一个数据中心。搜索相关性可让您减少长距离网络流量。位于 San Francisco 数据中心的搜索头仅从在 San Francisco 的节点获得结果，同时位于 Los Angeles 的搜索头仅从它们本地的节点获得结果。

搜索相关性如何工作

对于要支持搜索相关性的那些站点，您必须配置多站点群集化，这样站点有一组完整的可搜索数据和一个本地搜索头。搜索头位于任一特定站点，仅从其本地站点获得数据，只要该站点**有效**。

搜索相关性运行时，每个搜索头都会把其搜索发送到所有站点上的各对等节点，但仅本地对等节点会搜索数据（即节点的主要数据桶副本）并把结果返回至搜索头。

如果拥有部分主要可搜索数据的本地节点发生故障，而且站点暂时丧失有效状态，则在必要时，搜索头会在本地站点进行数据桶修复时从远程站点上的对等节点中获取数据。在此期间，搜索头仍然会从本地站点获取尽可能多的数据。

一旦站点重新获得其有效状态，新的搜索又会仅在本地站点之间进行。

更多有关群集如何处理搜索相关性的详细信息，请参阅“多站点索引器群集架构”和“在多站点群集中进行本地搜索”。

实现搜索相关性

对于多站点群集，默认启用搜索相关性。然而，您必须执行一些步骤来充分利用。具体地说，您必须确保可搜索数据和搜索头都能在本地获得。

要实现搜索相关性：

1. 配置站点搜索因子，这样在每个要求搜索相关性的站点上就有至少一份可搜索副本。

实现此操作的一种方式是为每个要求搜索相关性的站点明确指定搜索因子。例如，一个属性参数为 `site_search_factor = origin:1, site1:1, site2:1, total:3` 的四站点群集可确保 site1 和 site2 有每个数据桶的可搜索副本。第三组可搜索副本会被分散到两个非显式站点之间，无法保证任一站点上会有一组完整的可搜索副本。因此，仅在 site1 和 site2 上启用搜索相关性。site1 和 site2 每一个都将拥有所有数据桶的主要副本。

还有配置站点搜索因子的方法可确保即使没有明确指定某些或全部的搜索因子，所有站点上都有可搜索副本。例如，一个属性参数为 `site_search_factor = origin:1, total:3` 的三站点群集可确保每个站点有一个可搜索副本，并因此启用每个站点的搜索相关性。每一个站点都将拥有所有数据桶的主要副本。

关于复制和搜索因子如何在站点间分发副本的更多信息，请参阅“配置站点复制因子”和“配置站点搜索因子”。

2. 在需要搜索相关性的每个站点部署搜索头。

禁用搜索相关性

您可以对于任何搜索头禁用搜索相关性。当禁用搜索相关性时，搜索头不会尝试只从单个站点上获得搜索结果。反之，它会从多个站点上获得结果。例如，如果您有两个数据中心非常接近，相互间延迟很低，并且您希望通过在两个站点的索引器之间分散处理过程来提高整体性能，这会很有用。

当禁用搜索相关性时会发生什么

当在搜索头上禁用搜索相关性时，搜索结果可以来自任何或所有站点的索引器。如果站点搜索因子规定了在多个站点上的可搜索数据桶副本，则搜索头使用未定义的标准来选择要搜索哪个可搜索副本。它可能会从一个站点选择一些数据桶的副本，并从其他站点选择其他数据桶的副本，因此结果将来自多个站点。

搜索头总是从主要数据桶副本中选择。例如，假设您有两个使用该搜索因子的站点群集：

```
site_search_factor = origin:2, total:3
```

原始站点将为每个数据桶存储两个可搜索副本，而第二个站点将存储一个可搜索副本。因此，对于一些数据桶（在 site1 上生成的这些数据桶），site1 将有两个可搜索副本，而对于其他数据桶（在 site2 上生成的这些数据桶），site2 将有两个可搜索副本。然而，每个站点只有一个主要副本。

启用搜索相关性的搜索头会尽可能限制它的搜索在自己站点的主要副本中进行。

与此相反，禁用搜索相关性的搜索头将它的搜索分布在两个站点上的主要副本之间进行。对于一个给定的数据桶，您无法获知它是会选择在 site1 上的主要副本或是 site2 上的主要副本。从一个搜索到下一个搜索，它倾向于使用相同的主要副本。

如何禁用搜索相关性

如需禁用搜索头的搜索相关性，请在 server.conf 中把搜索头的站点值设置为 “site0”：

```
[general]
site = site0

[clustering]
multisite = true
...
```

设置 site=site0 后，搜索会因此表现得像在单个站点群集上，不会特别偏好任何站点。

有关配置多站点搜索头的更多信息，请参阅“配置搜索头”。

将主要副本限制在带搜索头的站点上

默认情况下，索引器群集会将主要数据桶副本限制在带搜索头的站点上。此行为大大减少了主要副本的修复和重新平衡活动，尤其是在所有搜索头都只分配给一个或几个站点且搜索相关性被禁用的情况下。

您可以在管理器节点的 server.conf 文件中使用设置 assign_primaries_to_all_sites 来更改此行为。默认情况下，该设置为 "false"，即群集会把主要副本限制在带搜索头的站点上：

```
[clustering]
assign_primaries_to_all_sites=false
```

当 assign_primaries_to_all_sites=false 时，管理器节点会把主要副本只分配给当前带在线搜索头的站点。例如，如果没有为 site1 分配在线搜索头，则不会将任何主要副本分配给 site1。同样，如果没有为 site0 分配搜索头，则不会将任何主要副本分配给 site0。

当搜索头加入以前没有分配搜索头的站点时，管理器节点会针对该站点上带可搜索副本的所有数据桶，将主要状态逐一分配给该站点上每个数据桶的每个可搜索副本。

如果某个站点失去了所有搜索头，则该站点上的所有现有主要副本都将保持原样，但管理器节点不会将主要状态分配给该站点上的任何新数据桶副本。

也可以在管理器节点上使用 CLI 命令来更改此设置：

```
splunk edit cluster-config -assign_primaries_to_all_sites false|true
```

使用 server.conf 配置多站点索引器群集

首先阅读

在阅读本主题前，请参阅：

- “多站点索引器群集部署概述”。本主题提供关于配置多站点群集的重要的背景信息。
- “使用 server.conf 配置索引器群集”。本主题介绍基本的群集配置。它侧重于单个站点群集，但是大多数信息也和多站点群集相关。

多站点配置和单个站点配置的差异

以和配置单个站点群集相似的方式配置多站点索引器群集，除了几个新的属性以外：

在所有多站点群集节点上（管理器节点/对等节点/搜索头）：

- site 属性指定节点驻留的站点。

在管理器节点和搜索头上：

- multisite 属性指示参与到多站点群集的管理器节点或搜索头。

仅在管理器节点上：

- available_sites 属性对管理器节点管理的站点命名。
- site_replication_factor 用单个站点群集代替所使用的标准 replication_factor。详细信息请参阅“配置站点复制因子”。
- site_search_factor 用单个站点群集代替所使用的标准 search_factor。有关详细信息，请参阅“配置站点搜索因子”。

重要提示：尽管 site_replication_factor 有效地取代了单站点 replication_factor，且 site_search_factor 也取代了单站点 search_factor，但这两个单站点属性会继续存在于管理器节点配置中，默认值分别为 3 和 2。只要有站点上的对等节点数量少于默认值，亦即如果任一站点上的对等节点数量只有一个或两个，则上述两个属性的存留会在启动时引发故障。此故障为一条消息，内容为多站点群集未满足其复制或搜索因子。例如，若某个站点只有两个对等节点，单站点复制因子默认值为 3 将引发故障。为避免或修复这个问题，您必须将单站点复制因子和搜索因子的值设置为低于或等于任意站点上对等节点的最低数量。为避免某个站点只有两个对等节点的情况，您必须将 replication_factor 属性的值明确设置为 2。请参阅“多站点群集不满足其复制或搜索因子”。

如果您正在将群集从单个站点迁移到多站点，则必须为现有的单个站点数据桶保留现有的 replication_factor 和 search_factor 属性，同时也要为新的多站点数据桶添加新的多站点属性 site_replication_factor 和 site_search_factor。请参阅“将索引器群集从单个站点迁移到多站点”。

配置多站点群集节点

要配置一个多站点群集，应为每个站点配置节点，编辑每个站点的 server.conf 文件。有关群集化属性的详细信息，请阅读 server.conf 规范。

站点值

站点值识别节点所驻留的站点。给多站点群集中的每个节点分配一个站点值。要执行此操作，应设置 site 属性（位于节点的 [general] 段落中）。

站点值的语法：

```
site<n>
```

其中 <n> 是一个范围从 0 到 63 的整数：site1，site2，site3，.....

例如：

```
site=site1
```

特殊值 "site0" 只能设置在搜索头或参与索引器发现的转发器上。

- 对于搜索头，"site0" 会禁用搜索头的**搜索相关性**。请参阅“禁用搜索相关性”。
- 对于参与索引器发现的转发器，"site0" 会使得转发器向所有站点上的所有对等节点发送数据。请参阅“在多站点群集中使用索引器发现”。

配置管理器节点

为整个管理器节点上的群集配置关键属性。下面是管理器节点的多站点配置示例：

```
[general]
site = site1

[clustering]
mode = manager
multisite = true
available_sites = site1,site2
site_replication_factor = origin:2,total:3
site_search_factor = origin:1,total:2
pass4SymmKey = whatever
```

```
cluster_label = cluster1
```

该示例指定了以下内容：

- 实例位于 site1。
- 实例为群集管理器节点。
- 群集是多站点的。
- 群集包含两个站点：site1 和 site2。
- 群集的复制因子为默认的 "origin:2,total:3"。
- 群集的搜索因子为 "origin:1,total:2"。
- 群集的安全密钥为 "whatever"。
- 群集标签为 "cluster1"。

请注意以下事项：

- 指定 site 属性（位于 [general] 段落中）。在 [clustering] 段落中指定所有其他多站点属性。
- 可在群集中查找任一站点的管理器节点，但是每个群集仅有一个管理器节点。
- 必须设置 multisite=true。
- 必须在 available_sites 属性中列出所有群集站点。
- 必须设置一个 site_replication_factor 和一个 site_search_factor。详细信息请参阅“配置站点复制因子”和“配置站点搜索因子”。
- 设置安全密钥的 pass4SymmKey 属性在所有群集节点上必须相同。详细信息请参阅“使用 server.conf 配置索引器群集”。
- 群集标签是可选项。群集标签对于识别监视控制台中的群集很有用。请参阅*《监视 Splunk Enterprise 手册》*中的“设置群集标签”。

重要提示：首次启动管理器节点时，它将阻止在对等节点上建立索引，直到您启用并重新启动个数等于全部复制因子的对等节点为止。例如，给出一个三站点群集，属性参数为 "site_replication_factor = origin:2, site1:1, site2:2, site3:3, total:8"，管理器节点将阻止建立索引直到在所有站点上总共至少有八个对等节点，包括至少是 site1 上一个，site2 上两个，site3 上三个。

当等待对等节点加入群集时不要重启管理器节点。如果重启了主节点，对等节点将需要再次重启。

配置对等节点

要在多站点群集中配置对等节点，应设置 site 属性（位于[general] 段落中）。在单个站点群集中，对节点的所有其他配置设置都完全相同。

下面是多站点对等节点的配置示例：

```
[general]
site = site1

[replication_port://9887]

[clustering]
manager_uri = https://10.152.31.202:8089
mode = peer
pass4SymmKey = whatever
```

该示例指定了以下内容：

- 实例位于 site1。一个对等节点只能属于一个单个站点。
- 此对等节点将使用端口 9887 来侦听从其他对等节点流入的复制的数据。您可指定任意可用但未使用的端口作为复制端口。请勿重复使用管理或接收端口。
- 此对等节点的群集管理器节点位于 10.152.31.202:8089。
- 实例为群集对等节点。
- 安全密钥为 "whatever"。

配置搜索头

多站点搜索头会提供搜索相关性。有关信息，请参阅“在多站点索引器群集中实现搜索相关性”。

要在多站点群集中配置搜索头，应设置 site 属性（位于 [general] 段落中）和 multisite 属性（位于 [clustering] 段落中）。在单个站点群集中，对于搜索头的所有其他配置设置都完全相同。下面是多站点搜索头节点的配置示例：

```
[general]
```

```
site = site1

[clustering]
multisite = true
manager_uri = https://10.152.31.202:8089
mode = searchhead
pass4SymmKey = whatever
```

该示例指定了以下内容：

- 实例位于 site1。每个群集的搜索头只能属于一个单个站点。
- 搜索头是多站点群集的一员。
- 此搜索头的群集管理器节点位于 10.152.31.202:8089。
- 实例为群集搜索头。
- 安全密钥为 "whatever"。

注意：也可使用 server.conf 来启用多群集搜索，其中搜索头在多群集、多站点或单个站点之间搜索。要在多个多站点群集之间搜索，可将搜索头配置成每个群集上不同站点的一员。有关详细信息，请参阅“为多站点群集配置多群集搜索”。

在重新配置一个已启动并正在运行的搜索头时，Splunk 推荐使用 CLI 命令（在“使用 CLI 配置多站点索引器群集”中有介绍），而不是直接编辑 server.conf。如果使用 CLI，您不需要重新启动搜索头。

重新启动群集节点

初始配置之后

将实例配置为多站点群集节点之后，需要重新启动所有节点（管理器节点、对等节点和搜索头）以使更改生效。您可以通过对每个节点调用 CLI restart 命令来实现此目的：

```
$SPLUNK_HOME/bin/splunk restart
```

重要提示：首次启动管理器节点时，它将阻止在对等节点上建立索引，直到您启用并重新启动个数等于全部复制因子的对等节点为止。例如，给出一个三站点群集，属性参数为 "site_replication_factor = origin:2, site1:1, site2:2, site3:3, total:8"，管理器节点将阻止建立索引直到在所有站点上总共至少有八个对等节点，包括至少是 site1 上一个，site2 上两个，site3 上三个。

当等待对等节点加入群集时不要重启管理器节点。如果重启了主节点，对等节点将需要再次重启。

更改配置之后

在管理器节点上

在更改了下列属性之后，必须重新启动管理器节点：

- multisite
- available_sites
- site_replication_factor
- site_search_factor

重新启动管理器节点后，还必须启动群集节点的滚动重新启动。如果没有做到，则群集将处于不确定状态。关于 splunk rolling-restart 命令的信息，请参阅“使用滚动重新启动”。

如果在管理器节点上更改了 site 值，则不需要重新启动。

在对等节点上

如果在对等节点上更改了 site 值，则必须重新启动以使更改生效。

重要提示：虽然最初启用一个实例作为群集对等节点时可以使用 CLI 命令 restart，但后续重新启动时不应使用此命令。复制开始后，restart 命令与索引复制不兼容。有关更多信息（包括安全重新启动方法的讨论），请参阅“重新启动单个对等节点”。

关于搜索头

如果是在搜索头上更改 site 值，则不需要重新启动。

使用 CLI 配置多站点索引器群集

首先阅读

在阅读本主题前，请参阅：

- “多站点索引器群集部署概述”。本主题提供关于配置多站点群集的重要的背景信息。
- “使用 CLI 配置索引器群集”。本主题介绍使用 CLI 配置群集的基础知识。它侧重于单个站点群集，但是大多数信息也和多站点群集相关。
- “使用 server.conf 配置多站点索引器群集”。本主题提供了关于配置多站点群集的有用信息，包括关于命令行选项（当前主题介绍的）对应属性的详细信息。

配置多站点群集节点

使用 `splunk edit cluster-config` 命令将实例配置为多站点群集节点。在启用某个实例之后，必须重新启动该实例。

站点值

站点值识别节点所驻留的站点。给多站点群集中的每个节点分配一个站点值。

站点值的语法：

```
>site<n>
```

其中 <n> 是一个范围从 1 到 63 的整数：site1，site2，site3，.....

注意：只有一个搜索头的情况下，您也可以把站点值设置为 "site0"。该设置会禁用搜索头的**搜索相关性**。

配置管理器节点

下面是管理器节点模式的多站点配置示例：

```
splunk edit cluster-config -mode manager -multisite true -available_sites site1,site2 -site site1 -site_replication_factor
origin:2,total:3 -site_search_factor origin:1,total:2 -secret your_key

splunk restart
```

该示例指定了以下内容：

- 实例为群集管理器节点。
- 群集是多站点的。
- 群集包含两个站点：site1 和 site2。
- 管理器节点位于 site1。
- 群集的复制因子为默认的 "origin:2,total:3"。
- 群集的搜索因子为 "origin:1,total:2"。
- 管理器节点，以及群集中的其他节点，将 "your_key" 用作其安全密钥。`-secret` 标记编辑 `server.conf` 的 `[clustering]` 段落中的 `pass4SymmKey` 设置。

请注意以下事项：

- 每个群集只有一个管理器节点。
- 必须将多站点群集管理器节点的 `multisite` 设置为 `true`。
- 必须使用 `available_sites` 属性列出所有群集站点。
- 必须设置一个 `site_replication_factor` 和一个 `site_search_factor`。详细信息请参阅“配置站点复制因子”和“配置站点搜索因子”。

您可能也需要调整单站点复制因子和搜索因子。请参阅“多站点配置和单站点配置的差异。”首次启动管理器节点时，它将阻止在对等节点上建立索引，直到您启用并重新启动个数等于全部复制因子的对等节点为止。例如，给出一个三站点群集，属性参数为 "site_replication_factor = origin:2, site1:1, site2:2, site3:3, total:8"，管理器节点将阻止建立索引直到在所有站点上总共至少有八个对等节点，包括至少是 site1 上一个，site2 上两个，site3 上三个。

当等待对等节点加入群集时不要重启管理器节点。如果重启了主节点，对等节点将需要再次重启。

如果您之后更改了管理器节点的 `site` 值，则不必重新启动管理器节点。

配置对等节点

要在多站点群集中配置对等节点，应设置 site 属性。在单个站点群集中，对节点的所有其他配置设置都完全相同。

下面是多站点对等节点的配置示例：

```
splunk edit cluster-config -mode peer -site site1 -manager_uri https://10.160.31.200:8089 -replication_port 9887 -secret
your_key

splunk restart
```

该示例指定了以下内容：

- 实例为群集对等节点。
- 实例位于 site1。一个对等节点只能属于一个单个站点。
- 此对等节点的群集管理器节点位于 10.160.31.200:8089。
- 此对等节点将使用端口 9887 来侦听从其他对等节点流入的复制的数据。您可指定任意可用但未使用的端口作为复制端口。请勿重复使用管理或接收端口。

如果您之后更改了对等节点的 site 值，则不必重新启动对等节点。

配置搜索头

要为多站点群集配置搜索头，应设置 site 参数。关于单个站点群集中的搜索头，所有其他设置都相同。

使用不同的命令对搜索头进行初始配置，之后再更改其配置。

初始配置搜索头：

使用 splunk edit cluster-config 命令。下面是多站点搜索头的配置示例：

```
splunk edit cluster-config -mode searchhead -site site1 -manager_uri https://10.160.31.200:8089 -secret your_key

splunk restart
```

该示例指定了以下内容：

- 实例为群集搜索头。
- 搜索头位于 site1。在每个群集中，一个搜索头只能属于一个站点。
- 此搜索头的索引器群集管理器节点位于 10.160.31.200:8089。

要禁用搜索头的搜索相关性，以便它能从群集中所有站点随机获取它的数据，可设置 site 属性为 "site0"。

注意：当指定 site 参数时，命令会自动设置 multisite=true（位于搜索头的 server.conf 文件中）。不需要显式传递 multisite 参数。

要之后再编辑搜索头配置：

使用 splunk edit cluster-master 命令，而不是 splunk edit cluster-config 命令。

例如，假定您使用 splunk edit cluster-config 命令初始配置一个单个站点搜索头：

```
splunk edit cluster-config -mode searchhead -master_uri https://10.160.31.200:8089

splunk restart
```

之后要为多站点群集重新配置搜索头，则使用 splunk edit cluster-master 命令：

```
splunk edit cluster-master https://10.160.31.200:8089  -site site1
```

重要提示：splunk edit cluster-master 命令总是把当前管理器节点的 URI:port 值作为其初始值。更多示例，请参阅“使用 CLI 配置索引器群集搜索头”。

关于为多群集搜索配置多站点搜索头的信息，请参阅“为多站点群集配置多群集搜索”。

注意：之后更改搜索头的 site 值不需要重新启动。

配置站点复制因子

首先阅读

在尝试配置站点复制因子之前，必须了解：

- 基本的、单个站点复制因子。请参阅“索引器群集架构的基础知识”和“复制因子”。
- 多站点群集配置。请参阅“使用 server.conf 配置多站点索引器群集”。

站点复制因子是什么

要实现多站点索引器群集化，必须配置站点复制因子。它代替了标准的复制因子，标准复制因子是单个站点部署所特有的。在管理器节点上指定站点复制因子，作为群集基本配置的一部分。

站点复制因子除了提供对整个群集中的副本总数进行控制之外，还提供对数据桶副本位置的站点级控制。例如，可指定一个两站点群集维护所有数据桶的共计三份副本，其中一个站点维护两份副本，第二个站点维护一份副本。

也可指定一个复制策略，该策略基于哪个站点生成数据桶。就是说，您可配置复制因子，以便接收外部数据的站点为源数据维护更大（与非源数据相比）数量的数据桶副本。例如，您可指定每个站点维护两份所有源数据的副本，但是仅有一份源数据副本在另一个站点。

语法

在管理器节点的 server.conf 文件中，使用 site_replication_factor 属性配置站点复制因子。该属性驻留在 [clustering] 段落，代替了单个站点的 replication_factor 属性。例如：

```
[clustering]
mode = manager
multisite=true
available_sites=site1,site2
site_replication_factor = origin:2,total:3
site_search_factor = origin:1,total:2
```

您还可以使用 CLI 来配置站点复制因子。请参阅“使用 CLI 配置多站点索引器群集”。

必须正确配置 site_replication_factor 属性。否则，管理器节点将不会启动。

出于灾难恢复的目的，请务必配置站点复制因子，以便群集在多个站点上维护每个数据桶的副本。

下面是正式的语法：

```
site_replication_factor = origin:<n>, [site1:<n>,] [site2:<n>,] ..., total:<n>
```

其中：

- <n> 是一个正整数，表明数据桶的副本数量。
- origin:<n> 指定一个将保留在站点上的数据桶副本的最小数量（此站点会在该数据桶中生成数据，即，数据首次进入群集的站点）。当一个站点生成数据时，它就被称为“源”站点。
- site1:<n>, site2:<n>, ..., 表明将保留在每个指定站点的副本的最小数量。标识符 "site1"、"site2" 等等，与在对等节点上指定的 site 属性值相同。
- total:<n> 指定每个数据桶的副本的总数，包括群集中的所有站点。

请注意以下事项：

- 此属性指定了按站点的复制策略。它是全局指定，应用到所有索引中的所有数据桶。
- 此属性仅在 mode=manager 和 multisite=true 的情况下有效。在那些情况下，它取代了所有 replication_factor 属性。
- 需要 origin 和 total 值。
- 站点值（site1:<n>, site2:<n>, ...）是可选项。在此处指定的站点被称为“显式”站点。没有指定的站点被称为“非显式”站点。
- 下面介绍群集如何决定站点获得的副本最小数量：
  - 当站点作为源站点时，站点获得的副本最小数量比它的任一站点值大（如果有的话），或为 origin。
  - 当站点不作为源站点时，则由站点值（如果有的话）决定站点获得的副本最小数量。
  - 非显式站点不保证有任何副本，除非它作为源站点工作。

例如，在一个四站点群集中（属性参数 "site_replication_factor = origin:2, site1:1, site2:2, site3:3, total:8"），site1 获得所有数据的两份副本，一份由其本身产生，一份由其他站点产生。Site2 获得两份数据副本，无论是不是它自己产生。Site3 获得三份数据副本，无论是不是它自己产生。非显式站点 site4 获得两份自己产生的数据副本，两份由 site2 或 site3 产生的数据副本，以及一份由 site1 产生的数据副本。（Site4 获得必需数量的副本，以确保每个数据桶的副本数量满足 total 值为 8 的要求。）下面根据数据来源计算 site4 的副本数量：

```
originate at site1 -> 2 copies in site1, 2 copies in site2, 3 copies in site3, 1 copy in site4 => total=8
originate at site2 -> 1 copy in site1, 2 copies in site2, 3 copies in site3, 2 copies in site4 => total=8
originate at site3 -> 1 copy in site1, 2 copies in site2, 3 copies in site3, 2 copies in site4 => total=8
originate at site4 -> 1 copy in site1, 2 copies in site2, 3 copies in site3, 2 copies in site4 => total=8
```

- 下面介绍在指定 site_replication_factor 时，如何计算所需的最小 total 值，且计算基于站点和 origin 值：
  - 如果有一些非显式站点，则 total 值必须最少是所有显式站点和 origin 值的和。

    例如，一个三站点群集（"site_replication_factor = origin:2, site1:1, site2:2"），total 必须至少为 5：2+1+2=5。对于另一个三站点群集（"site_replication_factor = origin:2, site1:1, site3:3"），total 必须至少为 6：2+1+3=6。

  - 如果所有站点都是显式的，则 total 必须至少为满足站点规定和 origin 值所需的最小值。

    例如，一个三站点群集（"site_replication_factor = origin:1, site1:1, site2:1, site3:1"），total 必须至少为 3，因为其配置要求永远不会超过三份副本。对于一个三站点群集（"site_replication_factor = origin:2, site1:1, site2:1, site3:1"），total 必须至少为 4，因为其中一个站点将一直作为源站点，这样就需要两份副本，同时其他每个站点仅需要一份副本。对于一个三站点群集（"site_replication_factor = origin:3, site1:1, site2:2", site3:3"），total 必须至少为 8，以包括 site1 作为源站点的情况。

    最简单的计算方法是将源的值代替最小的站点值，然后将站点值求和（包括被替代为源的值的那个站点）。所以，在最后一个示例中（"site_replication_factor = origin:3, site1:1, site2:2", site3:3"），site1 的值最小，是 1。将源的值 3 取代 1，然后加上 site2 和 site3 的值：3+2+3=8。

- 因为 total 值可能比一组显式值的总和大，群集需要有一个方案来处理“剩余的”数据桶副本。下面是处理方案：
  - 在所有站点值和源值都满足后，如果副本剩下待分配，则那些剩下的副本将在所有站点之间分布，副本少的或没有副本的站点优先，这样分布才会尽量均匀。假定有足够多的剩余副本可用，每个站点将拥有至少一份数据桶副本。

    例如，有一个四站点群集（"site_replication_factor = origin:1, site1:1, site4:2, total:4"），如果 site1 是源站点，则有一份剩余副本。此副本会被随机分布到 site2 或 site3。但是，如果 site2 是源站点，它获得一份副本，就没有剩余副本可分布到 site3。

    另一个示例，有一个四站点群集（"site_replication_factor = origin:2, site1:2, site4:2, total:7"），如果 site1 是源站点，则有三份剩余副本可分布。因为 site2 和 site3 没有显式分配的副本，三份副本在它们中间分布，每个站点至少可获得一份副本。但是，如果 site2 是源站点，它获得两份副本，site3 获得一份剩余的副本。

    整个过程取决于每个站点上足够数量的对等节点的可用性。如果一个站点没有足够的可用节点来接收其他副本，则此副本就会转到有可用节点的站点。在任何情况下，每个站点至少会分布或预留一份副本，假定有足够的副本可用。

    下面是更多示例：
    - 一个三站点群集，其 "origin:1, total:3"：保证每个站点分布一份副本。
    - 一个三站点群集，其 "origin:1, total:4"：保证每个站点分布一份副本，一份额外的副本转到至少有两个节点的站点上。
    - 一个三站点群集，其 "origin:1, total:9"，其中 site1 仅有一个节点，site2 和 site3 每个站点有 10 个节点：保证每个站点分布一份副本，剩下六份副本均匀地在 site2 和 site3 之间分布。
  - 如果一个非显式站点上的所有节点都有故障，且在所有其他非显式站点已收到一份副本之后还有剩余副本，则群集会为该站点预留一份剩下的副本，等待返回其节点。在此期间，site_replication_factor 无法满足，因为已分布的副本总数将比指定的 total 值小 1，原因是预留了一份副本给故障节点所在站点。

    例如，有一个三站点群集（"site_replication_factor = origin:1, site1:1, site4:2, total:5"），如果 site1 是源站点，将有两份剩余副本可在 site2 和 site3 之间分布。如果所有 site2 上的节点都有故障，则一份剩余副本转到 site3 ，另一份作为预留，直到 site2 中的一个节点重新加入群集。在此期间，site_replication_factor 无法满足。但是，如有一个四站点群集（"site_replication_factor = origin:1, site1:1, site4:2, total:4"，和上一个示例唯一的区别是 total 值是 4 而不是 5），如果 site1 是源站点，则只有一份剩余副本，会转到 site2 或 site3 上。如果所有 site2 上的节点都有故障，则副本会转到 site3，没有副本会预留给 site2。在此示例中会满足 site_replication_factor，因为没有副本预留给 site2。

- 每个站点必须部署一组节点，数量至少要和源值或其站点值中较大值一样大。

例如，有一个三站点群集（"site_replication_factor = origin:2, site1:1, site2:2, site3:3, total:7"），站点中

的节点数必须至少有：site1：2个节点；site2：2个节点；site3：3个节点。

- 部署在所有站点间的节点总数必须大于或等于 total 值。
- total 的值必须至少与 replication_factor 属性值（其默认值为 3）一样大。因此，如果 total 的值为 2，则您应该显式设置 replication_factor 的值为 2。
- 如果您正在从一个单个站点群集迁移，则单个站点群集的 total 值必须至少和 replication_factor 一样大。请参阅“将索引器群集从单个站点迁移到多站点”。
- 属性默认为："origin:2, total:3."

示例

- **一个两站点群集（site1，site2），其默认设置为 "site_replication_factor = origin:2, total:3"：**对于任何给定数据桶，源站点存储两份副本。其余的站点存储一份副本。
- **一个三站点群集（site1，site2，site3），其默认设置为 "site_replication_factor = origin:2, total:3"：**对于任何给定数据桶，源站点存储两份副本。两个非源站点中的一个被随机选中，存储一份副本，另一个不存储任何副本。
- **一个三站点群集（site1，site2，site3），其设置为 "site_replication_factor = origin:1, site1:1, site2:1, site3:2, total:5"：**对于所有的数据桶，site1 和 site2 每个站点存储最少一份副本，site3 存储两份副本。第五份副本分布到 site1 或 site2，因为这些站点所分配的副本比 site3 少。
- **一个三站点群集（site1，site2，site3），其设置为 "site_replication_factor = origin:2, site1:1, site2:1, total:4"：**Site1 存储两份自己生成的任一数据桶的副本和一份其他数据桶的副本。Site2 遵循同样的模式。Site3 的站点值没有明确定义，遵循同样的模式。
- **一个三站点群集（site1，site2，site3），其设置为 "site_replication_factor = origin:2, site1:1, site2:2, total:5"：**Site1 存储两份自己生成的任一数据桶的副本，一份或两份 site2 生成的任一数据桶副本，以及一份 site3 生成的任一数据桶的副本。Site2 存储两份任一数据桶的副本，无论是不是自己生成的。Site3 的站点值没有明确定义，存储自己生成的任一数据桶的两份副本，site1 生成的任一数据桶的一份副本，以及 site2 生成的任一数据桶的一份或两份副本。（当 site2 生成一个数据桶时，在初始分配后会保留一份副本。管理器节点将其随机分配给 site1 或 site3。）
- **一个三站点群集，设置为 "site_replication_factor = origin:1, total:4"：**每个数据桶的四份副本都在所有站点间随机分布，每个站点至少获得一份副本。

处理单站点复制因子的暂留

重要提示：尽管 site_replication_factor 有效地取代了单站点 replication_factor，但该单站点的属性会继续存在于管理器节点配置中，默认值为 3。只要有站点上的对等节点数量少于三个，上述属性的存留就会引发故障。此故障为一条消息，内容为多站点群集未满足其复制因子。例如，若某个站点只有两个对等节点，单站点复制因子默认值为 3 将引发故障。为避免或修复这个问题，您必须将单站点复制因子的值设置为低于或等于任意站点上对等节点的最低数量。为避免某个站点只有两个对等节点的情况，您必须将 replication_factor 属性的值明确设置为 2。请参阅“多站点群集不满足其复制或搜索因子”。

配置站点搜索因子

首先阅读

在尝试配置站点搜索因子之前，必须先了解：

- 基本的、单个站点搜索因子。请参阅“群集架构的基础知识”和“搜索因子”。
- 站点复制因子。请参阅“配置站点复制因子”。
- 多站点群集配置。请参阅“使用 server.conf 配置多站点索引器群集”。

站点搜索因子是什么？

要实现多站点索引器群集化，必须配置站点搜索因子。它代替了标准的搜索因子，标准搜索因子是单个站点部署所特有的。在管理器节点上指定站点搜索因子，作为群集基本配置的一部分。

站点搜索因子除了提供对整个群集中的可搜索副本总数进行控制之外，还提供对可搜索数据桶副本位置的站点级控制。例如，可指定一个两站点群集维护所有数据桶的共计三份可搜索副本，其中一个站点维护两份副本，第二个站点维护一份副本。

也可指定一个搜索策略，该策略基于哪个站点生成数据桶。就是说，您可配置搜索因子，以便接收外部数据的站点为源数据维护更大（与非源数据相比）数量的数据桶可搜索副本。例如，您可指定每个站点维护两份所有源数据的可搜索副本，但是仅有一份源数据副本在另一个站点。

站点搜索因子帮助确定群集是否有搜索相关性。请参阅“在多站点索引器群集中执行搜索相关性”。

语法

site_search_factor 和 site_replication_factor 的语法是一样的，除非有其他需求，如：site_search_factor 中的值和显式站点是 site_replication_factor 中的值和显式站点的子集。本部分详细介绍语法。

在管理器节点的 server.conf 文件中，使用 site_search_factor 属性配置站点搜索因子。该属性驻留在 [clustering] 段落，代替了单个站点的 search_factor 属性。例如：

```
[clustering]
mode = manager
multisite=true
available_sites=site1,site2
site_replication_factor = origin:2,total:3
site_search_factor = origin:1,total:2
```

您还可以使用 CLI 来配置站点搜索因子。请参阅“使用 CLI 配置多站点索引器群集”。

警告：必须正确配置 site_search_factor 属性。否则，管理器节点将不会启动。

下面是正式的语法：

```
site_search_factor = origin:<n>, [site1:<n>,] [site2:<n>,] ..., total:<n>
```

其中：

- <n> 是一个正整数，表明数据桶的可搜索副本数量。
- origin:<n> 指定一个将保留在站点上的数据桶可搜索副本的最小数量（此站点会在该数据桶中生成数据，即，数据首次进入群集的站点）。当一个站点生成数据时，它就被称为“源”站点。
- site1:<n>, site2:<n>, ..., 表明在每个指定站点保留的可搜索副本的最小数量。标识符 "site1"、"site2" 等等，与在对等节点上指定的 site 属性值相同。
- total:<n> 指定每个数据桶的可搜索副本的总数，包括群集中的所有站点。

请注意以下事项：

- 此属性指定了按站点的可搜索复制策略。它是全局指定，应用到所有索引中的所有数据桶。
- 此属性仅在 mode=manager 和 multisite=true 的情况下有效。在那些情况下，它取代了所有 search_factor 属性。
- 需要 origin 和 total 值。
- 站点值（site1:<n>, site2:<n>, ...）是可选项。在此处指定的站点被称为“显式”站点。没有指定的站点被称为“非显式”站点。
- 要确定一个站点获得的可搜索副本的最小数量，则和通过 site_replication_factor 确定一个站点获得的复制副本的最小数量使用相同的规则。请参阅“配置站点复制因子”。
- 要确定所需的最小 total 值，使用与确定最小 total 值（为 site_replication_factor 确定）一样的规则。请参阅“配置站点复制因子”。
- 因为 total 值可能比一组显式值的总和大，群集需要有一个方案来处理“剩余的”可搜索数据桶副本。此方案遵照为剩余的复制副本制定的方案，在“配置站点复制因子”中介绍。
- 所有的值必须小于或等于它们在 site_replication_factor 中相应的值。

  例如，有一个三站点群集（"site_replication_factor = origin:2, site1:1, site2:2, total:5"），那么在 site_search_factor 中，origin 值不能超过 2，site1 值不能超过 1，site2 值不能超过 2，且 total 值不能超过 5。
- 如果一个站点值已显式显示在 site_search_factor 中，则也必须显式显示在 site_replication_factor 中。但是，site_replication_factor 中的显式站点值并不要求也显式显示在 site_search_factor 中。

  例如，有一个三站点群集，设置为 "site_replication_factor = origin:2, site1:1, site2:2, total:5"（有一个非显式站点 site3），您可指定 "site_search_factor = origin:1, site2:2, total:4"（删除显式站点 site1），但是您不能指定 "site_search_factor = origin:1, site1:1, site2:2, site3:1, total:4"（将非显式站点 site3 设为显式）。
- 对于搜索相关性，必须配置 site_search_factor，这样在每个要求搜索相关性的站点上就有至少一份可搜索副本。只有显式

站点遵守搜索相关性。

- 如果您正在从一个单个站点群集迁移，则单个站点群集的 `total` 值必须至少和 `search_factor` 一样大。请参阅“将索引器群集从单个站点迁移到多站点”。
- 属性默认为："origin:1, total:2."

示例

关于站点搜索因子的语法示例，请参考“配置站点复制因子”中的示例。在 `site_search_factor` 中指定 `origin/site/total` 值的语法和 `site_replication_factor` 中的一样。

处理单站点搜索因子的暂留

重要提示：尽管 `site_search_factor` 有效地取代了单站点 `search_factor`，但该单站点的属性会继续存在于管理器节点配置中，默认值为 2。只要有站点仅含一个对等节点，上述属性的存留就会引发故障。此故障为一条消息，内容为多站点群集未满足其搜索因子。为避免或修复这个问题，您必须将单站点搜索因子的值设置为低于或等于任意站点上对等节点的最低数量。为避免某个站点只有一个对等节点的情况，您必须将 `search_factor` 属性的值明确设置为 1。请参阅“多站点群集不满足其复制或搜索因子”。

将索引器群集从单个站点迁移到多站点

您可将索引器群集从单个站点迁移到多站点。在此过程中，您可将现有的单站点群集合并到新的多站点群集中。

迁移后的数据桶行为

迁移后，迁移之前创建的所有数据桶会默认继续遵守单站点复制和搜索因子策略。您可以更改此行为，以使旧数据桶改为遵循多站点策略。

迁移后创建的数据桶始终遵守多站点策略。

将旧数据桶保留为单站点数据桶

默认情况下，迁移之后，群集同时保留单个站点和多站点的数据桶。将根据如下规则进行分别维护：

- 单个站点旧数据桶（那些在迁移时存在的）会继续遵守单个站点 `replication_factor` 和 `search_factor` 设置。
- 多站点数据桶（那些在迁移后创建的）会遵循多站点 `site_replication_factor` 和 `site_search_factor` 策略。

将旧数据桶转换为多站点数据桶

您可以配置管理器节点以将旧数据桶转换为多站点数据桶。此过程导致遵循单站点复制和搜索策略（预迁移）的数据桶改为遵循多站点复制和搜索策略。

在决定是否转换旧数据桶时，必须对比将这些数据桶转换为多站点数据桶所需的数据桶修复活动将耗费的可能相当大的时间成本，权衡将这些数据桶跨多个站点进行维护的价值。

您可以在迁移之前或迁移之后的任何时候进行必要的配置更改。

如果您在迁移之前更改配置，迁移后旧数据桶将立即遵守 `site_replication_factor` 和 `site_search_factor` 策略。

如果您在迁移之后更改配置，任何遵守单站点策略的迁移后创建的数据桶都将遵守多站点策略。

要查看有多少数据桶需要转换为多站点，请在更改管理器节点配置之前使用 `services/cluster/manager/buckets?filter=multisite_bucket=false&filter=standalone=false`。

配置管理器节点以将旧数据桶转换为多站点数据桶

要使得旧的单站点数据桶遵循多站点复制和搜索因子策略，请将管理器节点的 `server.conf` 文件中的 `constrain_singlesite_buckets` 设置更改为 "false"：

```
[clustering]
mode = manager
constrain_singlesite_buckets = false
```

必须重新启动管理器节点，更改才能生效。

执行多站点迁移

前提条件

- 管理器节点必须运行 Splunk Enterprise 7.2 或更高版本。
- 迁移后群集中的所有节点必须遵循 Splunk Enterprise 版本兼容性中介绍的版本兼容性规则。因此，在迁移到多站点前，您可能需要升级单个站点群集。遵循“升级索引器群集”中相应的过程。
- 如果您想要现有的数据桶在迁移后遵循多站点复制和搜索策略，您必须更改管理器节点上的配置。请参阅“配置管理器以将现有数据桶转换为多站点数据桶”。或者，您可以在迁移后随时执行此步骤。

步骤

要从单个站点群集迁移到多站点，应为多站点配置每个节点：

1. 遵循“使用 CLI 配置多站点索引器群集”中的说明，配置多站点的管理器节点并重启。例如：

```
splunk edit cluster-config -mode manager -multisite true -available_sites site1,site2 -site site1 -site_replication_factor
origin:2,total:3 -site_search_factor origin:1,total:2

splunk restart
```

请注意以下事项：

- 不要删除现有的复制因子和搜索因子的单个站点属性，`replication_factor` 和 `search_factor`。管理器节点需要它们来处理迁移的数据桶。
- `total` 值（`site_replication_factor` 和 `site_search_factor` 的值）必须至少分别与 `replication_factor` 和 `search_factor` 的值一样大。
- 如果任意站点的对等节点的数目少于单个站点（`replication_factor` 或 `search_factor`），则必须降低那些属性的值使之匹配任意站点对等节点的最小数目。例如，如果 `replication_factor` 是 3，`search_factor` 是 2，且有个站点仅有 2 个对等节点，您必须将 `replication_factor` 更改为 2。否则，迁移的数据桶可能不满足复制和搜索因子，这取决于群集复制迁移数据桶的方式。请参阅“多站点群集不满足其复制或搜索因子”。

2. 在管理器节点上设置维护模式：

```
splunk enable maintenance-mode
```

本步骤可以防止不必要的数据桶修复。请参阅“使用维护模式”。

要确认管理器节点进入了维护模式，运行 `splunk show maintenance-mode`。

3. 为多站点配置现有对等节点。为每个对等节点指定它的管理器节点和站点。例如：

```
splunk edit cluster-config -site site1
```

会出现重启对等节点的提示。

为每个对等节点执行此操作，指定它的站点。

4. 如果想要在群集中添加新的对等节点，请遵循“使用 CLI 配置多站点索引器群集”的说明。例如：

```
splunk edit cluster-config -mode peer -site site1 -manager_uri https://10.160.31.200:8089 -replication_port 9887

splunk restart
```

为每个想要添加到群集中的新对等节点执行此操作。

5. 为多站点配置搜索头。为每个搜索头指定它的管理器节点和站点。例如：

```
splunk edit cluster-master https://10.160.31.200:8089 -site site1
```

为每个搜索头执行此操作，指定它的站点。

注意：如果搜索头是一个搜索头群集的成员，则配置基本相同。请参阅*《分布式搜索》*中的“使用多站点索引器群集以集成”。

6. 如果想要在群集中添加新的搜索头，请遵循“使用 CLI 配置多站点索引器群集”的说明。例如：

```
splunk edit cluster-config -mode searchhead -site site1 -manager_uri https://10.160.31.200:8089

splunk restart
```

为每个想要添加到群集中的新搜索头执行此操作。

7. 在管理器节点上禁用维护模式：

```
splunk disable maintenance-mode
```

要确认管理器节点退出了维护模式，运行 `splunk show maintenance-mode`。

您可以查看管理器节点仪表板，以确认所有群集节点已经启用并正在运行。

在迁移过程中，群集用站点值来标记每个单个站点数据桶。

注意：您也可以通过直接编辑 `server.conf` 来配置多站点群集。请参阅“使用 server.conf 配置多站点索引器群集”

8. 如果您正使用索引器发现来连接转发器到对等节点，您必须为每个转发器分配一个站点。请参阅“在多站点群集中使用索引器发现”。

如果您已配置管理器节点将现有的单站点数据桶转换为遵守多站点复制和搜索因子策略，那么群集迁移进程完成后，数据桶修复可能会持续一段时间。如果您现有大量的数据桶，数据桶修复可能会花费很长时间。

群集如何迁移和维护现有的数据桶

多站点群集中的数据桶包含一个识别源站点的属性。单个站点群集中的数据桶不包含这一属性。所以，当群集从单个站点迁移到多站点时，它必须用一个源站点值来标记每个单个站点数据桶。因为数据桶名称包括原始对等节点的 GUID，所以群集始终知道原始节点。有了这一信息，群集可推断出数据桶的源站点：

- 如果原始对等节点仍然存在于群集中，则群集会假定数据桶来源于原始节点所被分配的站点。群集会将数据桶的来源设置为那个站点。
- 如果原始节点不再存在于群集中，则群集假定拥有最多数据桶副本的站点是源站点。群集会将数据桶的来源设置为那个站点。

如果将群集配置为将现有数据桶保留为单站点数据桶

这里说明群集如何使用推断源站点来继续维护单个站点的数据桶，如何处理一些必需的修复以使数据桶能继续满足单个站点的复制和搜索因子：

- 如果群集需要复制其他的数据桶副本来满足复制因子，它只在数据桶的推断的源站点内复制。
- 如果群集需要做一份可搜索数据桶的不可搜索副本来满足搜索因子，它可能会在非源站点上这样做，只要该数据桶的不可搜索副本已存在于一些其他站点。

群集绝不会在一个非源站点上创建数据桶的新副本。

因为，如果迁移后的数据桶保留为单站点数据桶，那么群集只会在初始站点上创建新的数据桶副本，这样可能会遇到不满足复制因子的情况。例如，如果 `replication_factor` 设为 3，但是初始站点上只有两个对等节点，这样群集就不会在非初始站点上创建第三个副本，即使非初始站点上之前可能存在数据桶的第三个副本。

要补救这种情况，您可将迁移前创建的数据桶转换为多站点数据桶，或者减少单站点复制因子数量，这样复制因子就不会超出任何站点上的对等节点数。

如果将群集配置为将现有数据桶转换为多站点数据桶

群集会使用介绍的方法推断出各数据桶的初始站点。

群集从遵循单站点策略转换为遵循多站点策略的过程和数据桶修复过程是一样的，包括跨站点流等。如果您现有大量的数据桶，这一过程可能会花费很长时间才能完成。修复过程的优先级和任何其他并行的修复过程是一样的。

如果继续修复，管理器节点仪表板会显示不满足复制因子和搜索因子。修复过程完成后，群集会返回到完成状态，如管理器节点仪表板所示。

处理新站点上受阻的索引

如果您为多站点群集化配置了管理器节点，但新站点尚未完全运行，则在管理器节点等待足够多的可用于满足多站点复制因子的对等节点时，它会阻止建立索引。要取消阻止索引，您可以在管理器节点上运行 `splunk set indexing-ready` 命令。请参阅“管理器节点重新启动或站点故障之后在多站点群集中重新启动建立索引”。

查看索引器群集状态

查看管理器节点仪表板

此仪表板提供有关整个索引器群集状态的详细信息。您还可以从中获得有关管理器节点的每个对等节点的信息。

有关其他群集化仪表板的信息，请参阅以下内容：

- “查看对等节点仪表板”
- “查看搜索头仪表板”

访问管理器节点仪表板

1. 单击 Splunk Web 右上角的**设置**。
2. 在**分布式环境**组中，单击**索引器群集化**。
   显示“管理器节点”仪表板。

只能在已作为管理器节点启用的实例上查看此仪表板。

查看管理器节点仪表板

管理器节点仪表板包含以下部分：

- 群集概述
- 对等节点选项卡
- 索引选项卡
- 搜索头选项卡

群集概述

群集概述汇总了群集的运行状况。它将告诉您以下信息：

- 群集数据是否完全可搜索；即，群集中的所有数据桶是否拥有**主要**副本。
- 是否满足搜索和复制因子。
- 多少个对等节点可搜索。
- 多少个索引可搜索。

根据群集的运行状况，它还可能提供如下警告消息：

- 一些数据不可搜索。
- 不满足复制因子。
- 不满足搜索因子。

有关群集概述显示的详细信息，请浏览下方的选项卡。

在仪表板右上角，这里有三个按钮：

- **编辑。**关于此按钮的信息，请参阅“使用仪表板配置管理器节点”。
- **更多信息。**本按钮提供有关管理器节点配置的详细信息：
  - **名称。**此管理器节点的 `serverName`（在管理器节点的 `$SPLUNK_HOME/etc/system/local/server.conf` 文件中指定）。
  - **复制因子。**群集的**复制因子**。
  - **搜索因子。**群集的**搜索因子**。
  - **生成期间 ID。**群集的当前**生成期间 ID**。
- **文档。**

对等节点选项卡

对于每个对等节点，管理器节点仪表板列出了以下信息：

- **对等节点名称。**此对等节点的 `serverName`（在对等节点的 `$SPLUNK_HOME/etc/system/local/server.conf` 文件中指定）。
- **完全可搜索。**本列指示节点当前是否有一组完整的主要副本，以及是否完全可搜索。
- **站点。**（仅针对多站点。）本列显示了每个节点的站点值。
- **状态。**对等节点的状态。有关此处所介绍过程的更多信息，请参阅“使对等节点脱机”。可能的值包括：
  - `Up`
  - `Pending`. 当复制失败时会出现这种情况。在对等节点下一次成功向管理器节点发送检测信号时，该值会转换回 `Up`。
  - `AutomaticDetention`. 对等节点会在磁盘空间不足时进入此状态。在此状态下，对等节点不会为外部或内部数据建立索

引，也不会用作复制目标。对等节点将继续参与搜索。请参阅“将对等节点置为滞留状态”。
    - ManualDetention. 对等节点通过手动干预进入此状态。在此状态下，对等节点不会获取外部数据或为其建立索引，也不会用作复制目标。对等节点将继续参与搜索。请参阅“将对等节点置为滞留状态”。
    - ManualDetention-PortsEnabled. 对等节点通过手动干预进入此状态。在此状态下，对等节点将继续获取外部数据并为其建立索引，但不会用作复制目标。对等节点将继续参与搜索。请参阅“将对等节点置为滞留状态”。
    - Restarting. 当您运行 splunk offline 命令（不带 enforce-counts 标记）时，该对等节点将在离开 ReassigningPrimaries 状态之后暂时进入此状态。它会在 restart_timeout 期间（默认值为 60 秒）内保持此状态。如果您没有在此时间内重新启动该对等节点，该对等节点之后将进入 Down 状态。在滚动重新启动期间或通过 Splunk Web 重新启动时，该对等节点也将进入此状态。
    - ShuttingDown. 管理器节点检测到该对等节点正在关闭。
    - ReassigningPrimaries. 当您运行 splunk offline 命令（不带 enforce-counts 标记）时，对等节点将暂时进入此状态。
    - Decommissioning. 当您运行 splunk offline 命令（带 enforce-counts 标记）时，该对等节点将进入并维持在此状态，直到所有数据桶修复完成且对等节点可以关闭。
    - GracefulShutdown. 当您运行 splunk offline 命令（带 enforce-counts 标记）时，该对等节点在取消配置状态成功结束并最终关闭时进入此状态。在脱机时该节点会一直处于该状态。
    - Stopped. 当您使用 splunk stop 命令停止该对等节点时，该节点会进入此状态。
    - Down. 出于 GracefulShutdown 或 Stopped 状态所导致的原因之外的原因，该对等节点在脱机时进入此状态：要么是您运行了 splunk offline 命令不带 enforce-counts 标记的版本且对等节点关闭的时间超过 restart_timeout 期间（默认值为 60 秒钟），或者是对等节点因其他某种原因（例如，对等节点崩溃）而脱机。
- **数据桶。**对等节点具有其副本的数据桶的数量。

要获得任何对等节点的更多信息，单击对等节点名称左侧的箭头。将显示这些字段：

- **位置。**对等节点的 IP 地址和端口号。
- **上一检测信号。**管理器节点从对等节点接收到最后一个检测信号的时间。
- **复制端口。**对等节点用来从其他对等节点接收复制的数据的端口。
- **基本生成期间 ID。**对等节点的基本生成期间 ID，这相当于群集在对等节点上次加入群集时的生成期间 ID。此 ID 将小于或等于群集的当前生成期间 ID。因此，如果一个对等节点在生成期间 1 加入群集，同时自此一直保持在群集中，则其基本生成期间 ID 保持为 1，即使群集可能增量其当前生成期间 ID，如 5。
- **GUID。**对等节点的 GUID。

注意：在一个对等节点关闭后，尽管它的状态更改为 "Down" 或 "GracefulShutdown"，但它会继续出现在对等节点列表中。要从管理器节点列表中删除对等节点，请参阅“从管理器节点列表中删除对等节点”。

索引选项卡

对于每个索引，管理器节点仪表板列出了以下信息：

- **索引名称。**索引的名称。内部索引前面带一个下划线 (_)。
- **完全可搜索。**该索引是否完全可搜索？换言之，该索引是否每个数据桶至少具有一个可搜索副本？如果索引中即使一个数据桶没有可搜索副本，此字段会将该索引报表为不可搜索。
- **可搜索的数据副本。**群集拥有的完整可搜索索引副本的数量。
- **复制的数据副本。**群集拥有的索引副本数量。每个副本必须完整，不得缺失任何数据桶。
- **数据桶。**索引中的数据桶数量。此数量不包括重复的数据桶副本。
- **累计原始数据大小。**索引的原始数据大小，不包括热数据桶。此数量不包括重复的原始数据副本。

索引列表包括内部索引 **_audit** 和 **_internal**。正如您在群集中所预期的那样，这些内部索引包含由群集中的*所有*对等节点生成的结合数据。如果您需要搜索某单一对等节点生成的数据，可以搜索该对等节点的主机名。

此选项卡也会显示一个带有**数据桶状态**标签的按钮。如果单击它，会转到数据桶状态仪表板。请参阅查看数据桶状态仪表板。

注意：新索引只在它包含一些数据之后才会出现在此处。换句话说，如果您在对等节点上配置了一个新索引，该索引的一行只在您向该索引发送数据后才会显示。

搜索头选项卡

对于访问本群集的每个搜索头，管理器节点仪表板列出：

- **搜索头名称。**此搜索头的 serverName（在其 $SPLUNK_HOME/etc/system/local/server.conf 文件中指定）。
- **站点。**（仅针对多站点。）本列显示了每个搜索头的站点值。
- **状态。**搜索头是打开还是关闭？当搜索头在两倍于 generation_poll_interval 的时间长度内均未轮询管理器节点以获得生成期间信息，则管理器节点判断此搜索头为关闭。该属性可在 server.conf 中配置。

注意：该列表将管理器节点包含为搜索头之一。尽管管理器节点具有搜索头功能，但是您应仅使用这些功能用于调试目的。管理器节点的资源必须专门用于满足其协调群集活动的关键角色。在任何情况下都不得将管理器节点部署为生产搜索头。同样，与专用搜索头不同，管理器节点上的搜索头不能被配置以用于多群集搜索；它只能搜索它自己的群集。

要获得任何搜索头的更多信息，单击搜索头名称左侧的箭头。将显示这些字段：

- **位置。**搜索头的服务器名称和端口号。
- **GUID。**搜索头的 GUID。

查看数据桶状态仪表板

数据桶状态仪表板提供了群集中数据桶的状态。包含三个选项卡：

- 修复任务 - 运行
- 修复任务 - 待定
- 过多的数据桶索引

修复任务 - 运行

该选项卡提供了当前正在修复的数据桶列表。例如，如果数据桶没有足够多的副本，要使群集回到**有效**和**完成**状态一定会发生修复活动。当那些活动发生的时候，在列表中就会出现数据桶。

修复任务 - 待定

该选项卡提供了正等着修复的数据桶列表。您可以通过搜索因子、复制因子和生成期间过滤修复任务。

更多有关数据桶修复活动的信息，请参阅对等节点关闭时的情况。

本选项卡还包括一个“操作”按钮，让您可以修复单个数据桶的问题。详细信息请参阅处理单个数据桶的问题。

过多的数据桶索引

该选项卡提供了过多的数据桶副本的索引列表。它枚举了有过多副本和过多可搜索副本的数据桶。它也枚举了每个类别中的总的过多副本。例如，如果索引 "new" 有一个带有三个过多副本的数据桶，其中一个副本是可搜索的，和一个带有一个过多副本的第二数据桶，其副本是不可搜索的，"new" 所在的那行就会显示：

- 带有过多副本的数据桶 2 个
- 带有过多可搜索副本的数据桶 1 个
- 过多副本共 4 个
- 过多可搜索副本共 1 个

要想从单个索引中删除过多副本，单击索引所在行的右侧的**删除**按钮。

要想从所有索引中删除过多副本，单击**删除所有过多数据桶**按钮。

有关过多数据桶副本的更多信息，请参阅“从索引器群集中删除多余的数据桶副本”。

使用监视控制台查看状态

您可以使用监视控制台来监视部署的大多数方面，包括索引器群集的状态。控制台上的可用信息复制了管理器节点仪表板上许多可用的信息。

更多信息请参阅使用监视控制台查看索引器群集状态。

查看对等节点仪表板

此索引器群集对等节点仪表板提供有关单个对等节点状态的详细信息。

有关群集中的所有对等节点上的信息的单个视图，使用管理器节点仪表板，如“查看管理器节点仪表板”所述。

访问对等节点仪表板

要查看对等节点仪表板：

1. 单击 Splunk Web 右上角的**设置**。

2. 在**分布式环境**组中，单击**索引器群集化**。

只能在已作为对等节点启用的实例上查看此仪表板。

查看仪表板

此仪表板提供有关对等节点状态的信息：

- **名称**。此对等节点的 serverName（在其 $SPLUNK_HOME/etc/system/local/server.conf 文件中指定）。
- **复制端口**。对等节点用来从其他对等节点接收复制的数据的端口。
- **管理器位置**。管理器节点的 IP 地址和端口号。
- **基本生成期间 ID**。对等节点的基本生成期间 ID，这相当于群集在**对等节点上次加入群集时**的生成期间 ID。此 ID 将小于或等于群集的当前生成期间 ID。因此，如果一个对等节点在生成期间 1 加入群集，同时自此一直保持在群集中，则其基本生成期间 ID 保持为 1，即使群集可能增量其当前生成期间 ID，如 5。

配置对等节点

对等节点仪表板右上侧的**编辑**按钮提供几个更改节点配置的选项。请参阅“使用仪表板配置对等节点”。

注意：编辑按钮对多站点群集来说是禁用的。

查看搜索头仪表板

此仪表板提供有关索引器群集中搜索头状态的详细信息。

访问仪表板

访问仪表板：

1. 单击 Splunk Web 右上角的**设置**。
2. 在**分布式环境**组中，单击**索引器群集化**。

只能在已经启用为群集搜索头的实例上查看此仪表板。

查看仪表板

仪表板列出搜索头属于的所有群集的管理器节点，以及每个群集状态的一些信息。

有关管理器节点及其群集的更多信息，单击每行左侧的箭头。

您可以选择仪表板右上角的**更多信息**按钮，获取有关搜索头本身的信息。

- **名称**。此搜索头的 serverName（在其 $SPLUNK_HOME/etc/system/local/server.conf 文件中指定）。

配置搜索头

仪表板提供几个在搜索头上操作或更改其配置的选项。请参阅“使用仪表板配置搜索头”。

查看有关搜索节点的信息

还可以在 Splunk Web 中从搜索头的“分布式搜索”页面中查看有关搜索头的**搜索节点**（在群集化中，与群集对等节点集相同）的信息：

1. 在搜索头上，单击 Splunk Web 右上角的**设置**。
2. 在**分布式环境**部分中，单击**分布式搜索**。
3. 单击**搜索节点**以查看搜索节点集。

警告：请不要使用 Splunk Web 中的“分布式搜索”页面来更改搜索头配置或添加对等节点。有关如何正确配置索引器群集搜索头的信息，请参阅“搜索头配置概述”。

使用监视控制台查看索引器群集的状态

您可以使用监视控制台监视部署的大多数方面。本主题介绍可用来深入了解索引性能的控制台仪表板。

监视控制台的主要文档位于*《监视 Splunk Enterprise 手册》*内。

在**索引**菜单下方有两个索引器群集化仪表板：

- 索引器群集化：状态
- 索引器群集化：服务活动

索引器群集化：“状态”仪表板提供有关您的群集状态的信息。在很大程度上，它复制了管理器节点仪表板中的信息，如“查看管理器节点仪表板”中所述。

索引器群集化：“服务活动”仪表板提供有关问题的信息，例如数据桶修复活动以及告警和错误。

有关更多信息，请查看仪表板本身。此外，请参阅“索引器群集化：状态”和“索引器群集化：服务活动”，该部分在*《分布式管理控制台手册》*中。

管理索引器群集

添加对等节点到群集

您可以随时将对等节点添加到索引器群集。若要这么做，您只需启用新的 Splunk Enterprise 实例作为对等节点。

在启用对等节点之前，请熟悉相关的系统要求，如在“索引器群集的系统要求和其他部署注意事项”中所述。确保 Splunk Enterprise 实例满足版本兼容性和所有其他记录在案的要求。

对于单站点群集，请采用以下其中一个操作：

- 要使用 Splunk Web 仪表板启用对等节点，请参阅“启用对等节点”。
- 要使用 CLI 启用对等节点，请参阅“使用 CLI 配置对等节点”。
- 要通过编辑 server.conf 启用对等节点，请参阅“使用 server.conf 配置对等节点”。

要为多站点群集启用对等节点，请参阅“使用 server.conf 配置多站点索引器群集”。

当对等节点启动时，管理器节点会将最新的配置软件包分发给对等节点。此过程确保新的对等节点与其他对等节点具有相同的配置集，包括索引集和其他索引设置。请参阅“对等节点启动时软件包的分发”。

启用对等节点后，您可能需要采取步骤以确保对等节点正在从转发器接收数据。有多种方法可以配置转发器和对等节点之间的关系。根据群集连接到转发器的方式，您可能需要配置新的对等节点或转发器，或同时配置两者。有关详细信息，请阅读“将数据导入索引器群集”一章中的相关主题。

最后，请考虑重新平衡群集的数据，以便将现有数据移到新的对等节点上。这让新的对等节点可以分担搜索负担。请参阅“重新平衡索引器群集数据”。

使对等节点脱机

使用 splunk offline 命令使对等节点脱机。

警告：请勿使用 splunk stop 使对等节点脱机。相反，请使用 splunk offline。它会以一种对搜索干扰最小的方式停止对等节点。

根据您的需求，您可以使对等节点永远或暂时脱机。在两种情况下，群集都会执行操作，以重新获得其**有效**和**完整**状态：

- **有效**群集具有其所有数据桶的**主要**副本，因此可以处理跨整个数据集的搜索请求。对于多站点群集的情况，有效群集的每个带有搜索相关性的站点同样有主要副本。
- **完整**群集具有与所有数据桶的**复制因子**相同的副本数，具有与**搜索因子**相同的可搜索副本数。因此，它符合故障容错的指定要求。完整群集还是有效群集。

脱机用例

splunk offline 命令有两个版本，一个版本是使对等节点暂时脱机，另一个则是使对等节点永远脱机。

当您使对等节点暂时脱机时

临时使对等节点脱机时，通常执行短期维护任务，例如计算机或操作系统升级。您希望该群集无中断地继续处理数据和搜索，但如果在对等节点脱机的这段时间内该群集不满足其复制因子或搜索因子，则也是可以接受的。

当对等节点暂时脱机时，管理器节点会启动操作让群集返回到有效状态，但通常不会让群集返回到完整状态，因为一旦对等节点重新联机，群集即可重新获得其完整状态。

当您使对等节点永远脱机时

当您使索引器群集对等节点永远脱机时，要确保该群集无中断地继续处理数据和搜索。也要求该群集替换因对等节点脱机而造成丢失的数据桶副本。例如，如果脱机对等节点保留了 10 个数据桶的副本（三个可搜索、七个不可搜索），则该群集必须在群集内的其他对等节点上重建这些副本，以满足其复制因子和搜索因子。

当对等节点永远脱机时，管理器节点会启动各种数据桶修复流程以使群集返回到有效和完整的状态。

Splunk 脱机命令

splunk offline 命令对两种类型的对等节点关闭情况都能处理：暂时的和永远的。该命令使对等节点正常关闭，尝试允许正在进行的搜索完成，同时也使群集快速恢复到有效状态。它试图以此种方式消除对现有或将来搜索的干扰。

splunk offline 命令还会启动补救数据桶修复活动，以使群集恢复到完整状态。它将立即启动此流程或在指定时间后再启动，让对等节点有时间重新联机并避免产生数据桶修复的需求，取决于您所运行命令的版本。

以下是与典型用例相对应的 splunk offline 命令的两个版本：

- splunk offline. 用于暂时停用对等节点以进行维护操作。也称为“快速版脱机”命令。
- splunk offline --enforce-counts. 用于从群集中永久删除对等节点。也称为“enforce-counts 版脱机”命令。

暂时停用对等节点：快速版的脱机命令

快速版 splunk offline 命令的简单语法如下：splunk offline.

群集试图在对等节点脱机之前恢复其有效状态。它不会试图恢复它的完整状态。可使用这个版本让对等节点暂时脱机，而无需启动任何数据桶修复活动。

当您想使对等节点永远脱机但必须即刻完成时，也可以使用此版本，只是在节点脱机后会发生数据桶修复活动。

快速版的脱机过程

在一定的约束条件下，对等节点在群集尝试满足两个条件之后脱机：

- 重新分配对等节点上的主要副本，以便群集恢复其有效状态
- 完成对等节点目前参与的所有搜索

在对等节点的主要数据桶副本重新分配给其他对等节点的可搜索副本之后，该对等节点会脱机以便群集恢复其有效状态。主要分配活动分配的最大时间期限由对等节点的 server.conf 文件中的 decommission_node_force_timeout 设置的值决定，默认为五分钟。

您还可以为 splunk offline 命令的单次执行更改主要分配超时期限。请参阅“快速版脱机命令的语法”。

注意：如果群集搜索因子为 1，则群集在允许对等节点脱机之前不会试图重新分配主要副本。如果搜索因子为 1，群集没有先创建新的可搜索副本则无法修复主要副本，这会花费大量的时间，因此无法实现快速关机的目的。

对等节点还会等待任何正在进行的搜索完成，直到最长时间，如 server.conf 中的 decommission_search_jobs_wait_secs 属性所确定的。此属性默认为三分钟。

一旦满足了这些条件，或者超出了活动的最大持续时间，对等节点就会脱机。

快速版脱机命令的语法

以下是快速版 splunk offline 命令的语法：

```
splunk offline
```

直接在对等节点上运行此命令。

当您运行此命令时，在群集返回到有效状态以及对等节点完成所有正在进行的搜索之后，对等节点关闭，如快速脱机过程中所述。

要更改群集分配的时间期限以返回到有效状态，请运行以下格式的命令：

```
splunk offline --decommission_node_force_timeout <seconds>
```

这将仅更改当前脱机操作的主要分配超时期限。例如，要将当前操作的超时期限更改为 10 分钟：

```
splunk offline --decommission_node_force_timeout 600
```

此对等节点的其他脱机操作将使用 server.conf 中保存的设置值，默认为五分钟。

在对等节点关闭之后，您有 60 秒的时间（默认情况）来完成维护工作，并通过调用 splunk restart 命令使对等节点恢复联机。如果对等节点在此时间内未恢复到群集，则管理器节点会启动数据桶修复活动，使群集恢复到完整状态。如果需要更多时间，则可配置 restart_timeout 属性，由此来延长管理器节点等待对等节点恢复联机的时间。详参“延长重新启动时间”中的介绍。

重要提示：要尽量减少所有数据桶的修复活动，通常应一次只停用一个对等节点。如果执行的操作涉及使许多对等节点暂时脱机，则考虑在运行期间调用维护模式。请参阅“使用维护模式”。

有关对等节点脱机时所发生进程的详细信息，请阅读“对等节点故障时的情况”。

延长重新启动时间

如果需要向对等节点执行维护，并且您预计所需的时间超出管理器节点的 restart_timeout 时间（默认设置为 60 秒），则可更改该设置的值。在管理器节点上运行以下 CLI 命令：

```
splunk edit cluster-config -restart_timeout <seconds>
```

例如，此命令将超时时间重置为 900 秒（15 分钟）：

```
splunk edit cluster-config -restart_timeout 900
```

您可动态运行此命令。运行此命令后，您不需重新启动管理器节点。

还可以在管理器节点的 server.conf 中更改此值。

永远停用对等节点：enforce-counts 版的脱机命令

脱机命令的 enforce-counts 版本用于仅在群集返回完整状态后使对等节点永远脱机。

在该版本的命令中，群集在允许对等节点脱机之前会执行必要的数据桶修复活动来恢复其有效和完整的状态。

警告：脱机命令的 enforce-counts 版本可确保脱机的对等节点的复制数据在群集中的其他对等节点上可用。但是，这不考虑或以任何方式处理可能存在于对等节点上的任何独立数据桶，因为节点可能较早地从非群集索引器迁移到群集，如“将非群集索引器迁移到群集环境中”所述。因此，如果您需要保留驻留在独立数据桶中的数据，您必须采取其它措施离开脱机过程。

enforce-counts 版的脱机过程

对等节点在群集满足两个条件之后脱机：

- 完成群集恢复其完整状态所必需的所有数据桶修复活动
- 完成对等节点正在参与的受时间限制约束的正在进行的搜索

仅在对等节点的可搜索和不可搜索数据桶副本重新分配给其他对等节点之后，该对等节点才会脱机以使群集恢复其完整状态。

因为此版本的 splunk offline 要求群集在对等节点脱机之前返回到完整状态，所以在运行此命令之前需要一定的先决条件：

- 群集必须有（复制因子 + 1）个对等节点，以便它可以根据需要将数据桶副本重新分配给其他对等节点，并且能在对等节点脱机后继续满足其复制因子。
- 在多站点群集中，从源或显式对等节点的数量方面来看，对等节点的站点必须有足够的对等节点以便站点继续满足站点复制因子的要求。
- 群集不能处于维护模式，因为在维护模式期间不会发生数据桶修复。

对等节点还会等待任何正在进行的搜索完成，直到最长时间，如 server.conf 中的 decommission_search_jobs_wait_secs 属性所确定的。此属性默认为三分钟。

enforce-counts 版脱机命令的语法

以下是 enforce-counts 版 splunk offline 命令的语法：

```
splunk offline --enforce-counts
```

直接在对等节点上运行此命令。

此命令版本启动一个名为**取消配置**的操作，操作期间管理器节点对大范围的补救过程进行协调。对等节点会在这些过程都完成且群集返回到完整状态后再关闭。如果对等节点保留了大量的数据桶副本的话，则可能需要一段时间才能完成。

返回完整状态实际所需的时间取决于对等节点所保留的数据类型和数据量。详细信息请参阅“评估对等节点取消配置时群集的恢复时间”。

如果群集无法返回完整状态，则对等节点无法关闭。这是由“enforce-counts 版的脱机过程”中所述的问题所导致。如果需要使一个对等节点脱机（虽然有这样的问题），则要改用快速版的 splunk offline 命令。

有关对等节点取消配置时所发生进程的详细信息，请阅读“对等节点故障时的情况”。

在一个对等节点关闭后，尽管它的状态变为 "GracefulShutdown"，但它会继续出现在管理器节点仪表板的对等节点列表中。要

从管理器节点列表中删除对等节点，请参阅“从管理器节点列表中删除对等节点”。

评估对等节点取消配置时群集的恢复时间

在使对等节点取消配置时，管理器节点会在剩余的对等节点之间协调活动，以修复数据桶，并使群集恢复到完整状态。例如，如果正在脱机的对等节点存储了 10 个数据桶副本，其中 5 个副本是可搜索副本，则管理器节点会指示对等节点：

- 将这 10 个数据桶副本以流化方式传送到其他对等节点，以使群集重新获得所有数据桶副本（以匹配复制因子）。
- 使 5 个不可搜索的数据桶副本成为可搜索副本，以使群集重新获得所有可搜索数据桶副本（以匹配搜索因子）。

此活动可能需要一些时间才能完成。究竟多长时间取决于许多因素，例如：

- **系统注意事项**（如 CPU 规范、存储类型、互连类型）。
- 创建可搜索数据桶的对等节点**当前所执行的其他索引量**。
- 在脱机节点上存储的**数据桶的大小和数量**。
- 在脱机对等节点上存储的可搜索副本中**索引文件的大小**。（这些索引文件大小相对于原始数据大小可能因分段量等因素而有极大不同。）有关原始数据和索引文件相对大小的信息，请参阅“存储注意事项”。
- **搜索因子**。这确定群集转换不可搜索副本为可搜索副本的快速程度。如果搜索因子最少为 2，群集可以通过复制可搜索副本的剩余集合到索引文件，将不可搜索副本转换为可搜索副本。然而，如果搜索因子为 1，则群集必须通过重新构建索引文件来转换不可搜索副本，这会花费更长时间。（有关数据桶中文件类型的信息，请参阅“数据文件”。）

尽管存在这些变化因素，您仍然可以粗略地确定进程需要的时间。假设您使用的是 Splunk Enterprise 参考硬件，下面是有关两个主要活动所需时间的一些基本估计值：

- 通过 LAN 以流化方式将 10GB（原始数据和/或索引文件）从一个对等节点传送到另一个对等节点大约需要 5-10 分钟。
- 重新构建包含 4GB 原始数据的不可搜索数据桶副本的索引文件所需的时间取决于一些因素，如由此产生的索引文件，但是 30 分钟是合理的所需时间。如果搜索因子是 1，需要重新构建索引文件，意味着没有索引文件的任何副本用于流化。包含 4GB 原始数据的不可搜索数据桶副本能在添加索引文件后增长至约 10GB。如前所述，实际大小取决于许多因素。

使用维护模式

维护模式停止了大部分的数据桶修复活动，并且阻止热数据桶的频繁滚动。当在索引器群集上执行对等节点升级和其他维护活动时，这很有用。因为它停止了关键的数据桶修复活动，所以只在必要时才使用维护模式。

为何使用维护模式

热数据桶复制期间，会生成某些条件，并导致数据来源对等节点滚动数据桶。尽管这一行为通常对于索引器群集的健康有好处，但是如果错误经常出现的话，这会导致整个群集出现许多小数据桶。会生成不可接受数量的小数据桶的情况包括持续的网络问题或对等节点重复脱机。

要停止这种行为，您可以临时将群集设置为维护模式。这对于生成重复网络错误的系统维护工作非常有用，如网络重新配置。类似，如果您需要升级对等节点或临时让几个对等节点脱机，则可调用维护模式以预先阻止数据桶在这段时间滚动。

注意：CLI 命令 `splunk apply cluster-bundle` 和 `splunk rolling-restart` 默认结合维护模式功能到它们的行为，因此无需在运行这些命令时显式调用维护模式。表明维护模式正在运行的消息会显示在管理器节点仪表板上。

维护模式对于群集操作的影响

为了阻止数据桶进行不必要地滚动，维护模式停止了大部分的数据桶修复活动。在维护模式期间出现的唯一数据桶修复是主要副本修复。管理器节点在必要时尝试重新分配主要副本给可用可搜索的数据桶副本。

特别是，群集不会执行需要将不可搜索的数据桶复制或转换为可搜索的数据桶的修复。这表示在维护模式期间，管理器节点不会强制执行复制因子或搜索因子策略。因此，如果群集在维护模式期间丢失了一个对等节点，则它可以在有效但不完整的状态下运行。要了解这种含义的信息，请参阅**索引器群集状态**。

同样，如果群集丢失的对等节点个数等于或大于复制因子，则在维护模式持续期间也会失去其有效状态。

此外，在维护模式中，如果群集失去哪怕一个对等节点，对于运行在主要副本修复后续期间的搜索，它可能会返回不完整的结果。这段时间通常很短，经常只有几秒钟，但是即使是主要副本修复的很短时间，仍然会影响正在进行的搜索。

维护模式的工作方式对于单个站点和多站点群集是相同的。它没有站点概念。

启用维护模式

在开始维护活动之前将群集设置为维护模式。一旦完成维护，您应禁用维护模式。

要调用维护模式，在管理器节点上运行本 CLI 命令：

```
splunk enable maintenance-mode
```

当运行 `enable` 命令时，将显示警告维护模式影响的消息，并需要您确认以便继续。

维护模式在管理器节点重新启动期间会持续，这在 6.6 版本上有效。

禁用维护模式

要返回标准数据桶滚动行为，运行：

```
splunk disable maintenance-mode
```

确定维护模式的状态

要确定是否打开维护模式，运行：

```
splunk show maintenance-mode
```

返回值 `1` 表示维护模式已打开。 `0` 指示维护模式已关闭。

重新启动整个索引器群集或单个对等节点

本主题介绍了如何重新启动整个索引器群集（不常见）或单个对等节点。

重新启动管理器节点或对等节点时，管理器节点将重新平衡跨一组对等节点的主要数据桶副本，详参“重新平衡索引器群集的主要数据桶”中的介绍。

有关需要重新启动的配置更改的信息，请参阅“修改 server.conf 后重新启动”和“配置软件包更改后重新启动或重新加载”。

重新启动整个群集

通常不需要重新启动整个群集。如果更改了管理器节点配置，只要重启管理器节点。如果更新了一组通用的对等节点配置，管理器节点只重新启动这组对等节点，而且只会在必要时才重新启动。详参“更新通用对等节点配置”中的介绍。

如果您出于任何原因确实需要同时重新启动管理器节点和对等节点：

1. 重新启动管理器节点，方法和任何实例一样。例如，在管理器节点上运行以下 CLI 命令：

```
splunk restart
```

2. 管理器节点重新启动后，请等待所有对等节点在管理器节点重新注册，并且管理器节点仪表板指示所有对等节点和索引均可搜索。请参阅“查看管理器节点仪表板”。

3. 在管理器节点上运行以下 CLI 命令，作为一个组重新启动对等节点：

```
splunk rolling-restart cluster-peers
```

请参阅“执行索引器群集的滚动重启”。

如果需要重新启动搜索头，可以随时执行此操作，只要群集的其余部分处于运行状态即可。

重新启动单个对等节点

您可能偶尔需要重新启动单个对等节点；例如，您只在该对等节点上更改某些配置时。

请勿使用 CLI `splunk restart` 命令重新启动对等节点，原因请参阅本节后文中的介绍。相反，您可以使用两种方法安全地重新启动单个对等节点：

- 使用 Splunk Web（**设置 > 服务器控制**）。
- 先运行 `splunk offline` 命令，然后再运行 `splunk start`。

您使用 Splunk Web 或 `splunk offline`/`splunk start` 命令重新启动对等节点时，管理器节点会在假设对等节点已经永久故障前等待 60 秒（默认情况）。这样，对等节点便可以有充足的时间恢复联机，从而防止群集执行不必要的补救活动。

注意：管理器节点等待的实际时间取决于 server.conf 中管理器节点的 restart_timeout 属性值。此属性默认为 60 秒。如果需要管理器节点等待更长的时间，可以更改 restart_timeout 值，详参“延长重新启动时间”中的介绍。

splunk offline/splunk start 重新启动方法优于 Splunk Web 方法之处在于，在停止对等节点之前，重新启动方法会等待正在进行的搜索完成。另外，因为重新启动方法包括两个步骤过程，所以在您执行一些维护的过程中需要对等节点短暂地保持关闭状态时，可以使用此方法。

有关 splunk offline 命令的信息，请参阅“使对等节点脱机”。

警告：请勿使用 splunk restart 命令重新启动对等节点。如果使用 splunk restart 命令，管理器节点将不会注意到对等节点正在重新启动。相反，在默认等待对等节点发送检测信号 60 秒后，管理器节点将启动在对等节点故障通常采取的补救操作，例如，将其数据桶副本添加到其他对等节点。管理器节点等待的实际时间由管理器节点的 heartbeat_timeout 属性决定。未经咨询，建议您不要更改 60 秒的默认值。

执行索引器群集的滚动重启

滚动重新启动将执行所有对等节点的分阶段重新启动，以便整个索引器群集可以在重新启动过程期间继续执行其功能。滚动重新启动还有助于确保向群集发送数据的负载均衡转发器始终有对等节点可以接收数据。

在以下情况下会发生滚动重新启动：

- 启动 Splunk Web 中的滚动重新启动。
- 您可以运行 splunk rolling-restart CLI 命令。
- 在将配置软件包分发到对等节点后，管理器节点会在必要时自动启动滚动重新启动。有关该进程的详细信息请参阅“分发配置软件包”。

滚动重新启动模式

索引器群集有两种滚动重新启动模式：

- **滚动重新启动：**在连续分组中重新启动对等节点（根据预先定义的百分比），但不保证群集是可搜索的。请参阅“滚动重新启动如何工作”。
- **可搜索的滚动重新启动：**重新启动对等节点时每次启动一个，尽量减少正在进行的搜索中断情况。请参阅“执行可搜索的滚动重启”。

要为 splunk rolling-restart cluster-peers 命令设置默认滚动重新启动模式，请参阅“在 server.conf 中设置滚动重新启动行为”。

滚动重新启动如何工作

在滚动重新启动期间，大约 10%（默认）的对等节点会同时进行重新启动，直到群集中的所有对等节点完成重新启动。如果群集中对等节点少于 10 个，则每次有一个对等节点进行重新启动。管理器节点安排重新启动过程，在每个对等节点轮到重新启动时发消息给对等节点。

重新启动百分比告诉管理器节点在滚动重新启动过程中多少个重新启动时隙要保持打开状态。例如，如果群集有 30 个对等节点，重新启动百分比设置为默认值 10%，则管理器节点保持三个时隙打开以供对等节点重新启动。当滚动重新启动过程开始时，管理器节点会向三个对等节点发出重新启动消息。一旦每个对等节点完成了重新启动并与管理器节点联系之后，管理器节点会向另一个对等节点发出重新启动消息，依此类推，直到所有对等节点都已重新启动。正常情况下，在该示例中，将始终有三个对等节点正在重新启动，直到过程结束。

如果由于机器配置不足或其他原因，对等节点重新启动缓慢，则同时进行重新启动的对等节点的数量会超过重新启动百分比。请参阅“处理缓慢重启”。

在滚动重新启动期结束后，管理器节点重新平衡群集的主要数据桶。请参阅“重新平衡索引器群集的主要数据桶”以深入了解此过程。

以下是关于滚动重新启动的行为要注意的几件事情：

- 管理器节点以随机的顺序重新启动对等节点。
- 在滚动重新启动期间，群集进入维护模式。在对等节点进行重新启动时，这可以防止不必要的**数据桶修复**。
- 在滚动重新启动期间，无法保证群集将完全可搜索。

启动滚动重新启动

您可以从 Splunk Web 或命令行启动滚动重新启动。

从 Splunk Web 启动滚动重新启动

1. 登录到管理器节点实例。
2. 单击**设置** > **索引器群集化**。
3. 单击**编辑** > **滚动重新启动**。

4. （可选）在“重新启动的对等节点百分比”字段中，输入数字以更改您想要同时重新启动的管理器节点的对等节点百分比。默认百分比为 10。

   如果您更改百分比，管理器节点会覆盖 `server.conf` 中 `percent_peers_to_restart` 的默认值，然后新值会变为默认值。

5. （可选）如果群集是一个多站点群集，您可以更改群集中站点重新启动的顺序。请勾选**指定站点顺序**复选框，然后单击下拉框，按照您想要重新启动的顺序安排可用站点。

   只有多站点群集才会显示站点顺序下拉框。

6. 单击**开始滚动重新启动**。

从命令行启动滚动重新启动

您可以从管理器节点调用 `splunk rolling-restart` 命令：

```
splunk rolling-restart cluster-peers
```

指定一次要重新启动的对等节点的百分比

默认情况下，一次重新启动 10% 的对等节点。重新启动百分比可以通过 `percent_peers_to_restart` 属性（位于 `[clustering]` 段落，在 `server.conf` 中）配置。为方便起见，您可以使用 CLI 命令 `splunk edit cluster-config` 配置本设置。

例如，要使得 20% 的对等节点同时重新启动，运行本命令：

```
splunk edit cluster-config -percent_peers_to_restart 20
```

要使得所有对等节点立即重新启动，运行命令时采用值 100：

```
splunk edit cluster-config -percent_peers_to_restart 100
```

立即重新启动所有对等节点在某些情况下有用，如没有用户正在主动搜索，同时没有转发器正在主动发送数据给群集。这可最大化减少完成重新启动所需的时间。

在更改 `percent_peers_to_restart` 值后，您必须运行 `splunk rolling-restart` 命令以真正开始重新启动。

执行可搜索的滚动重启

Splunk Enterprise 7.1.0 及更高版本可为滚动重新启动提供可搜索的选项。可搜索选项使您可以执行对等节点滚动重新启动，尽量减少正在进行的搜索中断情况。当由于定期维护或配置软件包推送需要进行滚动重启时，您可以使用可搜索的滚动重新启动尽量减少搜索中断。

可搜索的滚动重新启动如何工作

当您启动可搜索的滚动重新启动时，管理器节点会同时重新启动所有对等节点。每个对等节点重新启动过程中，管理器节点会将数据桶主要副本重新分配到其他对等节点，以保持可搜索状态，对等节点会在可配置超时值内完成任何正在进行的搜索。然后，管理器节点会重新启动对等节点，对等节点会重新加入群集。对每个对等节点重复此过程，直到滚动重新启动完成。

以下是关于可搜索的滚动重新启动行为需要注意的几个事项：

- 管理器节点每次重新启动一个对等节点。
- 在启动可搜索的滚动重新启动之前，管理器节点会运行运行状况检查，以确定群集处于可搜索状态。
- 对等节点会等待正在进行的搜索完成，直到最长时间，如 `server.conf` 中的 `decommission_search_jobs_wait_secs` 属性所确定的。此属性默认为 180 秒。此适用于大多数情况下的大部分搜索。
- 可搜索的滚动重新启动适用于历史搜索和实时搜索。
- 在索引器群集的可搜索滚动重启期间，默认情况下，所有连续的计划搜索和实时计划搜索都将暂停，直到重启完成为止。

可搜索滚动重启的最佳做法

在启动可搜索的滚动重启之前，为了确保最终取得成功，请考虑采用以下最佳做法：

- 为确保索引器不会陷入重新分配的管理器节点状态，并且能够在可搜索的滚动重新启动期间重新启动，请在管理器节点的 `server.conf` 的 `[clustering]` 段落中设置以下属性：

  ```
  [clustering]
  restart_timeout = 600
  rolling_restart = searchable_force
  decommission_force_timeout = 180
  ```

  关于以上属性的更多信息，请参阅*《管理员手册》*中的 `server.conf`。

  如果从 CLI 或 Splunk Web 启动滚动重启时指定了 `searchable` 或 `searchable_force` 选项，则指定选项的优先级将高于 `server.conf` 中现有的 `rolling_restart` 设置。

- 搜索时间超出 `decommission_search_jobs_wait_secs`（默认为 180 秒）的正在进行的搜索可能产生不完整的结果和响应的错误消息。如果有必须完成的计划搜索，请增加 `server.conf` 中 `decommission_search_jobs_wait_secs` 属性的值，或在搜索期间不要进行可搜索的滚动重启。
- 确保 `limits.conf` 的 `[search]` 段落中的 `search_retry` 属性已设为 `false`（默认）。将此属性设为 `true` 可能导致搜索时间超过 `decommission_search_jobs_wait_secs`，产生重复或部分结果，但没有错误信息。

启动可搜索的滚动重新启动

您可以从 Splunk Web 或命令行启动可搜索的滚动重新启动。

从 Splunk Web 启动可搜索的滚动重新启动

1. 在管理器节点上，单击**设置 > 索引器群集**。
2. 单击**编辑 > 滚动重新启动**。
3. 在**索引群集滚动重新启动**模型中，选择**可搜索**。

4. 单击**开始滚动重新启动**。
   此操作将启动可搜索的滚动重新启动。
5. （可选）尽管运行状况检查失败，要进行可搜索的滚动重新启动，请选择**强制**选项。此选项会覆盖运行状况检查，并允许继续进行可搜索的滚动重新启动。

使用强制选项时需谨慎。此选项可能会影响搜索。

6. 单击**开始滚动重新启动**。
   此操作将启动可搜索的滚动重新启动。您可以通过管理器节点仪表板监视可搜索的滚动重新启动进度。

从命令行启动可搜索的滚动重新启动

要从命令行执行可搜索的滚动重新启动：

1. （可选）运行初级运行状况检查以确定群集是否处于可搜索状态（满足搜索因子，且所有数据可搜索）。
2. 启动可搜索的滚动重新启动（包括运行状况检查）。
   还可使用强制选项启动可搜索的滚动重新启动（覆盖运行状况检查）。

1. （可选）运行初级运行状况检查

要检查群集的当前运行状况，请在管理器节点上运行以下命令：

```
splunk show cluster-status --verbose
```

此命令显示关于群集状态的信息。启动可搜索的滚动重新启动之前，查看命令输出以确认群集处于可搜索状态（满足搜索因子，所有数据均可搜索）。

群集必须有两份每个处于可搜索状态的数据桶的可搜索副本才能进行可搜索的滚动重新启动。

此示例为 `splunk show cluster-status --verbose` 命令输出的内容：

```
splunk@manager1:~/bin$ ./splunk show cluster-status --verbose

Pre-flight check successful .................. YES
 ├──────── Replication factor met ............... YES
 ├──────── Search factor met .................... YES
 ├──────── All data is searchable ............... YES
 ├──────── All peers are up ..................... YES
 ├──────── CM version is compatible ............. YES
 ├──────── No fixup tasks in progress ........... YES
 └──────── Splunk version peer count { 7.1.0: 3 }

 Indexing Ready YES

 idx1          0026D1C6-4DDB-429E-8EC6-772C5B4F1DB5        default
         Searchable YES
         Status  Up
         Bucket Count=14
         Splunk Version=7.1.0

 idx3          31E6BE71-20E1-4F1C-8693-BEF482375A3F        default
         Searchable YES
         Status  Up
         Bucket Count=14
         Splunk Version=7.1.0

 idx2          81E52D67-6AC6-4C5B-A528-4CD5FEF08009        default
         Searchable YES
         Status  Up
         Bucket Count=14
         Splunk Version=7.1.0
```

输出表明运行状况检查成功，即表示群集处于可搜索状态，可进行可搜索的滚动重新启动。

运行状况检查输出详细信息

此表显示用于确定索引器群集运行状况的条件的输出值。

运行状况检查	输出值	描述
满足复制因子	是	群集有指定数量的原始数据副本。
满足搜索因子	是	群集有指定数量的数据可搜索副本。
所有数据均可搜索	是	群集有所有数据的可搜索副本。
CM 版本兼容	是	管理器节点正在运行可兼容的 Splunk Enterprise。
没有正在进行的修复任务	是	没有正在进行的群集补救活动（例如数据桶复制或索引不可搜索的数据桶副本）。
所有对等节点已启用	是	所有索引器群集对等节点正在运行。
Splunk 版本对等节点计数	7.1 或更高版本 ：对等节点数量	运行 Splunk Enterprise 版本的对等节点数量：

运行状况检查并不全面。检查只适用于所列出的条件。

2. 启动可搜索的滚动重新启动

要执行可搜索的滚动重启：

在管理器节点上，使用 `searchable` 选项调用 `splunk rolling-restart cluster-peers` 命令。

```
splunk rolling-restart cluster-peers -searchable true
```

此命令会自动针对群集进行运行状况检查。如果运行状况检查失败，则此命令会返回以下消息，表明群集不处于可搜索状态。

```
"Request rejected. Wait until search factor is met and all data is searchable."
```

如果在运行状况检查失败的情况下，您想要继续可搜索的滚动重新启动，请使用 force 选项覆盖运行状况检查，然后启动可搜索的滚动重新启动，操作方法如下所示：

在管理器节点上，使用 force 选项调用 splunk rolling-restart cluster-peers 命令。

```
splunk rolling-restart cluster-peers -searchable true \
-force true \
-restart_inactivity_timeout <secs> \
-decommission_force_timeout <secs>
```

使用 force 选项后，您可以为以下其他参数指定自定义值。如果您没有指定参数，则使用其默认值。

- decommission_force_timeout：管理器节点强制重新启动对等节点后的时间（以秒为单位）。默认值：180.
- restart_inactivity_timeout：管理器节点认为对等节点重新启动失败并继续重新启动其他对等节点的时间（以秒为单位）。默认值：600.

在 server.conf 中设置滚动重新启动默认行为

您可以使用 server.conf 的 rolling_restart 属性（位于 [clustering] 段落中）设置滚动重新启动默认行为。此属性使您可以定义管理器节点在对等节点上执行的滚动重新启动类型。还可以提供一种便捷的方式为 splunk rolling-restart cluster-peers 命令自动加载选项，以替代从命令行传递选项。

rolling_restart 属性支持以下设置：

- restart：启动滚动重新启动。
- shutdown：启动阶段性滚动重新启动。关闭单个对等节点，然后等待手动重新启动。重复此过程，直到重新启动所有对等节点。
- searchable：在搜索干扰最小的情况下执行滚动重启。
- searchable_force：覆盖群集运行状况检查，然后启动可搜索的滚动重新启动。

指定 CLI 或 UI 中的 searchable 选项会覆盖 rolling_restart = shutdown 设置，位于 server.conf.

要设置 rolling_restart 属性：

1. 在管理器节点上，编辑 $SPLUNK_HOME/etc/system/local/server.conf。
2. 在 [clustering] 段落中，指定 rolling_restart 属性值。例如：

   ```
   [clustering]
   mode = manager
   replication_factor = 3
   search_factor = 2
   pass4SymmKey = whatever
   rolling_restart = searchable
   ```
3. 重新启动管理器。

设置 rolling_restart = searchable_force 后，您可以为 [clustering] 段落中的以下其他属性指定自定义值。如果您没有指定属性，则使用其默认值。

- decommission_force_timeout：管理器节点强制重新启动对等节点后的时间（以秒为单位）。默认值：180.
- restart_inactivity_timeout ：管理器节点认为对等节点重新启动失败并继续重新启动其他对等节点的时间（以秒为单位）。默认值：600.

有关更多信息，请参阅*管理员手册*中的 server.conf.spec。

为软件包推送将可搜索滚动重新启动设为默认模式

需要重新启动使用 server.conf 中 rolling_restart 值的配置软件包推送。您可将 rolling_restart 值设为 searchable，以为软件包推送触发的所有滚动重新启动将可搜索的滚动重新启动设为默认模式。

要为配置软件包推送将可搜索的滚动重新启动设为默认模式，指定以下其中一个 server.conf 属性（位于 [clustering] 段落）：

rolling_restart = searchable 或 rolling_restart = searchable_force

关于索引器群集的配置软件包推送的更多信息，请参阅“将软件包应用到对等节点”。

禁用延迟的计划搜索

默认情况下，可搜索滚动重新启动期间，根据 savedsearches.conf 中的 defer_scheduled_searchable_idxc 属性，连续的计划搜索会延迟，直到完成重新启动。无论如何设置，实时计划搜索会延迟。您可禁用此默认行为，这样连续的计划搜索就不会延迟，如下所示：

1. 在搜索头上，编辑 $SPLUNK_HOME/etc/system/local/savedsearches.conf。
2. 将 defer_scheduled_searchable_idxc 设置为 false。

   ```
   [default]
   defer_scheduled_searchable_idxc = false
   ```
3. 重新启动 Splunk。

禁用 defer_scheduled_searchable_idxc 之后，已保存的计划搜索可能会返回部分结果。

要重新启用延迟的计划搜索，请设置 defer_scheduled_searchable_idxc = true。

有关 defer_scheduled_searchable_idxc 的更多信息，请参阅*管理员手册*中的 savedsearches.conf。

在多站点群集上滚动重新启动

在多站点群集中，默认情况下，滚动重新启动会进行站点识别。即，管理器节点会先在一个站点上重新启动所有的对等节点，然后才在下一个站点上继续重新启动对等节点，依此类推。这确保了群集一直是完全可搜索的，假定每个站点都有一个完整的主要副本集的话。

在多站点群集上调用滚动重新启动

当您在多站点群集上调用 splunk rolling-restart 命令时，在继续下一个站点的对等节点之前，管理器节点会完成这个站点上所有对等节点的滚动重新启动。

通过 -site-order 参数，您可以指定站点重新启动的顺序。

以下是命令的多站点版本：

```
splunk rolling-restart cluster-peers [-site-order site<n>,site<n>, ...]
```

请注意以下有关 -site-order 参数的要点：

- 该参数指定站点重新启动的顺序。
- 您可指定站点子集。按照指定顺序仅重新启动指定的站点。
- 默认情况下，如果未指定该参数，则随机选择站点。

例如，有一个三站点群集，您可以使用此命令指定滚动重新启动：

```
splunk rolling-restart cluster-peers -site-order site1,site3,site2
```

管理器节点按以下顺序：site1、site3、site2 来执行重新启动。因此，管理器节点首先在 site1 的对等节点上执行滚动重新启动并一直等到 site1 的对等节点完成重新启动。然后管理器节点在 site3 上执行滚动重新启动并一直等到它完成。最后，在 site2 上执行滚动重新启动。

如果不希望对等节点按站点逐个重新启动，而宁愿让管理器节点从所有站点中随机选择下一个重新启动的对等节点，请使用参数 -site-by-site=false。

管理器节点如何确定每轮中要重新启动的多站点对等节点的数量

多站点集群的每个站点中可以同时重新启动的对等点数由 server.conf 设置决定：

- searchable_rolling_site_down_policy。默认值是一半。
- percent_peers_to_restart。默认为 10%。

应用 searchable_rolling_site_down_policy 设置之前，多站点群集的每个站点中必须有一个可搜索副本。此外，群集必须是健康的，对等节点不可以有站点限制的数据桶。如果满足条件，站点会使用 searchable_rolling_site_down_policy 设置决定重新启动的对等节点的数量。

如果未满足任何条件，进程会使用 percent_peers_to_restart 计算允许同时重新启动的对等节点数量。

重新启动的运行方式是：

1. 设置重新启动策略，代表站点中可以同时重新启动的对等节点数量。
2. 首先管理器选择一个要重新启动的站点。站点顺序是可配置的。

3. 管理器开始从第一个站点重新启动对等节点。不会跨多个站点重新启动对等节点。
4. 选定的对等点变为 ReassigningPrimaries 状态，然后是 Restarting、BatchAdding，最终状态是 Up。
5. 管理器会继续从第一个站点重新启动对等节点，直到所有站点对等节点都重新启动。
    1. 如果对等节点遇到问题，如进行滚动重新启动进程时超时，策略和对等节点计数不会更改。管理器会继续，直到站点中的每个对等节点均已重新启动。
6. 管理器开始在下一个站点上重新启动对等节点。
7. 管理器会继续重新启动下一个站点中的对等节点，一旦站点完成，会开始重新启动另一个站点上的对等节点，直到所有站点中的所有对等节点均已重新启动。

处理缓慢重启

如果对等节点实例重新启动缓慢，当管理器节点通知下一组开始重新启动时，这组中的对等节点可能仍在重新启动。这种情况可能会发生，例如，由于计算机资源不足。要解决此问题，您可以增加 restart_timeout 的值（位于管理器节点的 server.conf 文件中）。默认值为 60 秒。

冲突操作

您不能同时运行某些操作：

- 数据重新平衡
- 删除多余的数据桶
- 滚动重新启动
- 滚动升级

如果您触发其中一项操作，同时另一项操作也正在运行，那么 splunkd.log、CLI 和 Splunk Web 都会遇到问题，大意是当前存在冲突操作。

重新平衡索引器群集

通过重新平衡索引器群集，您可以平衡对等节点集的数据桶副本的分布。一组平衡的数据桶副本会优化每个对等节点的搜索负载，以及在数据重新平衡的情况下，优化每个对等节点的磁盘存储。

索引器群集重新平衡的类型

索引器群集重新平衡分两种：

- 主要副本重新平衡
- 数据重新平衡

主要副本重新平衡

主要副本重新平衡的目标是平衡对等节点间的搜索负载。

主要副本重新平衡在对等节点集上重新分布主要数据桶副本。它会尝试尽可能确保每个对等节点上的主要副本数量大致相同。

主要副本重新平衡只是简单地把主要标记重新分配给现有的可搜索副本集。它并不会把可搜索副本移动到不同的对等节点上。因为这个限制，主要副本重新平衡不可能实现主要副本的完美平衡。

因为主要重新平衡只是重新分配标记且不会导致任何数据桶副本在对等节点之间移动，所以进行得很快。

请参阅“重新平衡索引器群集的主要数据桶副本”。

数据重新平衡

数据重新平衡的目标是平衡对等节点间的存储分布。

数据重新平衡会重新分布数据桶副本，让每个对等节点上的副本数量大致相同。它将平衡可搜索、不可搜索的副本和主要副本。

数据重新平衡期间，群集把数据桶副本从副本较多的对等节点移动到副本较少的对等节点。由于重新平衡的类型包括可搜索副本，所以数据重新平衡将克服主要副本重新平衡的固有限制，并在平衡主要副本时取得明显更好的结果。

由于数据重新平衡涉及大量的修复活动，例如在对等节点之间移动数据桶副本，因此这是一个缓慢而漫长的过程。

请参阅“重新平衡索引器群集数据”。

重新平衡索引器群集的主要数据桶副本

当启动或重新启动管理器节点或对等节点时，管理器节点将重新平衡跨一组对等节点的主数据桶副本，以尝试尽可能公平地扩展主要副本。理想情况下，如果您有四个对等节点和 300 个数据桶，每个对等节点将保留 75 个主要副本。主要副本重新平衡的目的是均衡一组对等节点上的搜索负载。

主要副本重新平衡如何工作

为实现主要副本重新平衡，管理器节点将在必要时，把主要状态从现有数据桶副本重新分配给其他对等节点上相同数据桶的**可搜索**副本。本重新平衡是最好的尝试；无法保证提供完全、完美的重新平衡。

只要对等节点或管理器节点加入或重新加入群集，主要副本重新平衡就会自动发生。在滚动重新启动的情况下，在流程结束的时候，会出现一次重新平衡。

即使新的对等节点加入群集时会发生主要副本重新平衡，该节点也不会参与重新平衡，因为它还没有任何数据桶副本。重新平衡发生在所有现有的节点之间，这些节点要有可搜索的数据桶副本。

在数据桶上执行主要副本重新平衡时，管理器节点仅会将主要状态从一个可搜索副本重新分配到相同数据桶的另一个可搜索副本。如果有含可搜索副本的数据桶，执行主要副本重新平衡将改善所有对等节点上主要副本的平衡。这不会导致对等节点流化数据桶副本，同时不导致对等节点使不可搜索副本变为可搜索。如果现有对等节点没有任何可搜索副本，它将不会在重新平衡期间获得任何主要副本。

手动启动主要副本重新平衡

如果希望手动启动主要副本重新平衡进程，您可以重新启动对等节点或点击管理器节点上的 `/services/cluster/manager/control/control/rebalance_primaries` REST 端点。例如，在管理器节点上运行本命令：

```
curl -k -u admin:pass --request POST \
  https://localhost:8089/services/cluster/manager/control/control/rebalance_primaries
```

有关详细信息，请参阅*《REST API 参考手册》*中的 cluster/manager/control/control/rebalance_primaries。

重新平衡多站点群集上的主要副本

主要副本重新平衡在多站点群集中的工作方式存在一些差别。在多站点群集中，多个站点通常有完整的主要副本集合。当重新平衡群集时，对每个站点来讲，重新平衡是独立进行的。例如，在一个两站点群集中，群集分别重新平衡 site1 和 site2 中的主要副本。它不会在两个站点之间转移主要副本。

在任意站点上启动或重新启动对等节点将触发所有站点上的主要副本重新平衡。例如，在一个两站点群集中，在 site1 上重新启动一个节点，重新平衡会在 site1 和 site2 上发生。

查看对等节点上的主要副本数量

为了解任意对等节点上的主要负载，您可以使用 cluster/manager/peers 端点来查看某对等节点当前持有的主要副本数量。primary_count 显示对等节点的本地站点所持有的主要副本数量。primary_count_remote 显示对等节点的本地站点所持有的主要副本数量，包括 site0。

通过在所有对等节点上使用本端点，您可以确定群集是否可以受益于主要副本重新平衡。

请参阅*《REST API 参考手册》*中的 cluster/manager/peers。

索引器群集主要副本重新平衡的摘要

主要副本重新平衡是对群集中现有可搜索副本上主要分配的重新平衡。

在这些情况下，将出现主要副本重新平衡：

- 对等节点加入或重新加入群集
- 在滚动重新启动结束时
- 管理器节点重新加入群集
- 您手动调用管理器节点上的 rebalance_primaries REST 端点

重新平衡索引器群集数据

为重新平衡索引器群集数据，您需要重新平衡数据桶副本组，让每个对等节点所持有的副本数量大致相同。该操作可帮您确保每个对等节点都拥有近似的存储分布。

不平衡数据的问题

如果数据分布不平衡，一个或多个对等节点耗尽磁盘空间并由此转换到滞留状态的可能性会增加。在滞留状态下，对等节点不再为新数据建立索引，转发器因此也不会再发送数据到该节点。转发器停止发送数据后，负载均衡转发器会把传入的数据移转到其他对等节点上，加重这些节点在建立索引时的负担。最糟糕的情况下，如果转发器未配置负载均衡功能，则该转发器会丢失数据。

此外，随着对等节点上现有的数据逐渐老化，处于滞留状态下的对等节点中所包含的数据与其他对等节点相比，会相对较旧。由于大多数搜索会注重于较新的数据，这意味着该对等节点上的数据被搜索到的频率通常较低，由此把搜索负载的负担移转到未处于滞留状态中的对等节点上。

除滞留方面的考量外，不平衡数据也可能会影响搜索过程中对等节点的利用率。就某一特定索引而言，如果一些对等节点持有的数据桶副本比其他对等节点多，则在该索引上执行搜索时，这些节点会负担较大的搜索工作负荷。因为这个原因，数据重新平衡会识别索引器。数据重新平衡结束后，每个对等节点上各索引的数据桶副本数量会大致相同。

导致不平衡数据的条件

一些因素可能会造成数据桶副本的分布失衡。包括以下内容：

- **新添加了对等节点。**新添加对等节点时，里面最初是没有数据桶副本的。通过数据重新平衡，您可以把副本从其他对等节点上移动到新添加的节点。
- **数据转发不均衡。**如果转发器发送到某些对等节点上的数据较多，这些节点持有的数据桶副本可能会较多。重新平衡提供了一种方法，让您可以把副本从这些对等节点移动到副本较少的对等节点。

数据重新平衡完成了什么

数据重新平衡尝试实现对等节点集的数据桶副本的均衡分布。它将几个数据桶特性中的因素包括在内：

- 数据重新平衡将平衡不可搜索和可搜索的数据桶副本。
- 数据重新平衡为每个索引器平衡数据分布，因此，除了每个对等节点持有的数据桶副本总量大致相同外，每个对等节点上各索引的数据桶副本数量也相同。请参阅“数据重新平衡和索引”。
- 数据重新平衡的操作仅限于温数据桶和冷数据桶。不会重新平衡热数据桶。
- 数据重新平衡操作仅限于满足复制因子和搜索因子的数据桶。

数据重新平衡属于尽最大努力而为的平衡尝试，并非最完美的平衡尝试。请参阅“配置数据重新平衡的阈值”。

数据重新平衡和存储利用率

数据重新平衡对其对存储利用率的影响存在限制。

数据重新平衡进程平衡的是数据桶副本的数量，而非实际的数据存储。此外，数据重新平衡尝试实现的是实际平衡，而非完美平衡。大多数情况下，该进程会实现存储的最佳近似平衡，具体由以下几方面决定：

- 该进程假设所有对等节点就各可用索引而言，拥有的磁盘存储量都相同。最佳做法是在各对等节点上使用同类实例。
- 尽管数据桶大小偶尔会有差异，但该进程假设所有数据桶大小均相同。
- 每个对等节点上的副本数量都落入完美平衡的狭小范围内之后，进程将停止。该进程通常不会尝试在每个对等节点上都分布完全相同的副本数量。请参阅“配置数据重新平衡的阈值”。

确保数据重新平衡搜索安全

您可以在搜索模式中运行数据重新平衡以避免搜索停机时间。在可搜索模式中，重新平衡操作的搜索是安全的，具体取决于通过超时设置可控制的限制。

请勿将可搜索数据重新平衡与 SmartStore 索引一起使用。可搜索模式不是 SmartStore 的最佳选择，可能会造成数据重新平衡进程缓慢。改用不可搜索的数据重新平衡。

任何情况下，SmartStore 索引的非可搜索数据重新平衡通常只会导致最小程度的搜索中断。SmartStore 索引上数据重新平衡过程运行非常快速，因为此过程只移动数据桶的元数据，并不是数据桶数据本身。

数据重新平衡的默认模式是非可搜索模式，这意味着此操作的搜索并非安全。因为，作为其操作的一部分，在将副本流式传输到新对等节点之后，数据重新平衡会删除旧的对等节点中的数据桶副本，在搜索过程中，数据重新平衡操作可能会删除搜索需要的数据桶副本。因此，不保证非搜索数据重新平衡过程中的搜索结果是完整的。

使用可搜索数据重新平衡时，操作将等到正在进行的搜索完成后再开始，但不超过可配置的时间限制，然后再删除与搜索生成相关的旧数据桶副本。有较新生成内容的新搜索可同时对新的数据桶副本进行搜索。

在可搜索模式中完成数据重新平衡可能需要更长时间。可搜索数据重新平衡还需要更大的存储空间，因为会批量删除超出的数据桶，而不是立即删除，非可搜索数据重新平衡是立即删除。

最佳做法是在运行可搜索的数据重新平衡之前，单独删除多余的数据桶。请参阅“从索引器群集中删除多余的数据桶副本”。

通过单独删除多余的数据桶，删除流程作为不可搜索的操作运行，这比可搜索的删除操作（在可搜索的数据重新平衡开始之前发生）快得多。性能提高对于大量的多余数据桶（上千个或更多）来说尤为明显。

启动数据重新平衡操作时，您可以通过 CLI（如“启动数据重新平衡”中所述）或管理器节点仪表板（如“使用管理器节点仪表板启动和配置重新平衡”中所述）根据各操作指定可搜索模式。

您还可以通过管理器节点上的 server.conf 中的 searchable_rebalance 设置为所有数据重新平衡操作指定可搜索模式。如果您设置 searchable_rebalance = true，每次启动数据重新平衡时都不需要指定可搜索模式。

如果您发现您的一些搜索在数据重新平衡时不完整，您可以更改 server.conf 中的 rebalance_search_completion_timeout 设置来增加数据重新平衡操作等待正在进行的搜索完成的最大时间。默认值为 180 秒。

未重试到完成时间超出 rebalance_search_completion_timeout 的搜索。

必要时，server.conf 中的其他设置允许您控制可搜索数据重新平衡操作的其他方面。咨询 Splunk 支持之后才能修改这些设置。

可搜索数据重新平衡要求管理器节点和所有对等节点运行 Splunk Enterprise 7.3.0 或更高版本。

数据重新平衡如何工作

管理器节点控制着数据重新平衡进程。为实现在所有对等节点上均衡分布数据桶副本的目标，管理器节点会把数据桶副本从副本数量高于平均值的对等节点移动到副本数量低于平均值的对等节点。主节点会继续数据重新平衡进程，直到群集上的数据实现平衡为止，即每个对等节点持有的数据桶副本数量大致相同。

注意：当您启动数据重新平衡操作时，群集会先删除多余的数据桶，以删除超出群集复制因子或搜索因子的任何数据桶副本。请参阅“从索引器群集中删除多余的数据桶副本”。如下所述，群集还会删除数据重新平衡操作期间的多余数据桶，以清除操作期间创建的多余数据桶。

要实现重新平衡，群集会使用数据桶修复的基本过程。它将数据桶副本从一个对等节点流式传输到另外一个对等节点。因此，将数据桶副本“移动”到其他对等节点会将数据桶副本数量超出平均数量的对等节点中的副本流式传输到数据桶副本数量低于平均数量的对等节点中。

群集按顺序逐个处理数据桶，直到重新平衡完成为止。主节点不会等到一个数据桶完成重新平衡后才开始处理另一个，所以通常会有很多数据桶同时进行重新平衡处理。请参阅“控制重新平衡负载”。

如果是非可搜索数据重新平衡，将数据桶流式传输到其他对等节点之后，群集会立即删除整体数据桶副本数量超出平均数量的对等节点中的数据桶副本，来处理当前存在的多余的数据桶副本。群集在数据重新平衡的过程完成后执行主要数据重新平衡。

如果是可搜索数据重新平衡，在任何活跃的搜索完成之后，群集会批量删除多余的副本。通过批量删除多余的副本并等待活跃的搜索完成，群集能够尽量避免仍在运行搜索时删除正在进行搜索的数据桶。此外，要容纳正在进行的搜索以及新搜索，在批量删除多余的副本之前，群集还会切换主要数据。此操作会产生新的生成内容。正在进行的搜索会继续使用旧的生成内容，新搜索会使用新的生成内容。这样，无论何时开始搜索，整个过程中重新平衡操作的搜索都是安全的。

重新平衡进程可以提前终止，原因可能是手动干预也可能是时间限制到了。终止条件详参本主题中的其他地方。

数据重新平衡和索引

重新平衡进程按索引平衡数据桶副本。重新平衡完成后，每个对等节点持有的数据桶副本总数大致相同，且各索引分配到的副本数量也相同。

例如，假设您有一个群集，上面有四个对等节点和两个索引，分别为 index1 和 index2。index1 上有 100 个数据桶副本分布在所有对等节点上。index2 上有 300 个副本分布在所有对等节点上；两个索引上共计有 400 个副本分布在所有对等节点上。

重新平衡之前，对等节点组上的数据桶分布可能如下：

- Peer1：共计 110 个
  - Index1：10
  - Index2：100
- Peer2：共计 100 个
  - Index1：50
  - Index2：50
- Peer3：共计 50 个
  - Index1：20
  - Index2：30
- Peer4：共计 140 个
  - Index1：20
  - Index2：120

重新平衡后，数据桶分布大致如下：

- Peer1：共计 100 个
  - Index1：25
  - Index2：75
- Peer2：共计 100 个
  - Index1：25
  - Index2：75
- Peer3：共计 100 个
  - Index1：25
  - Index2：75
- Peer4：共计 100 个
  - Index1：25
  - Index2：75

多站点索引器群集中的数据重新平衡

多站点数据重新平衡的主要操作与单个站点重新平衡的操作相同。然而，在多站点群集中，管理器节点会在站点配置允许的情况下，首先平衡站点间的每个数据桶。之后，主节点才会在每个站点内平衡数据桶。

您在各站点上平衡多站点群集的力度取决于站点的复制因子和搜索因子。例如，若您的站点复制因子为 `origin:2,total:3`，群集将把三分之二的副本保留在它们的源站点上。如果某站点生成的数据桶数量超过其他站点，会导致数据失衡，而重新平衡无法在遵照站点复制因子的情况下解决此问题，因此，索引数据失衡的问题将继续存在。类似地，群集也不会在违反显式站点要求的情况下执行重新平衡。不过，站点间平衡确实会平衡各站点上的非显式副本。

启动数据重新平衡

您可以为所有索引或单个索引重新平衡数据。此外，您还可以为重新平衡设置一个时间限制。

如需重新平衡数据，请在管理器节点上运行以下 CLI 命令：

```
splunk rebalance cluster-data -action start [-searchable true] [-index index_name] [-max_runtime interval_in_minutes]
```

请注意以下事项：

- 将可选 `-searchable` 参数设为 `true` 以启用搜索安全的数据重新平衡。在可搜索模式中完成数据重新平衡可能需要更长时间。此参数默认为 `false`（非可搜索数据重新平衡）。
- 若仅平衡单个索引，请使用可选的 `-index` 参数。否则，CLI 命令将重新平衡所有索引。
- 使用可选的 `-max_runtime` 参数限制重新平衡活动的时长（以分钟为单位）。到达设定的时间限制后，即使还有数据桶要处理，重新平衡进程也会自动停止。数据重新平衡提前停止时会发生什么状况？详细信息请参阅“停止数据重新平衡”。

您也可以从管理器节点仪表板上启动重新平衡。请参阅“使用管理器节点仪表板启动和配置重新平衡”。

最佳做法是在维护窗口中执行数据重新平衡，原因如下：

- 数据重新平衡可以让主要数据桶副本移动到新的对等节点，所以数据重新平衡尚在进行期间无法保障搜索结果的完整。
- 和数据重新平衡相关的修复活动与其他数据桶修复活动（如维护复制因子和搜索因子）相比，优先级较低，所以重新平衡会等到其他修复活动完成之后才会开始。

停止数据重新平衡

如需提前停止数据重新平衡，请在管理器节点上运行以下 CLI 命令：

```
splunk rebalance cluster-data -action stop
```

运行此命令时如有任何数据桶正处于重新平衡进程中，群集将完成当前的进程。但是，群集不会在数据桶上启动任何其他处理进程。例如，若群集当时正在复制数据桶的不可搜索副本，群集会完成此复制过程，但不会继续处理可搜索副本，因此也不会检查数据桶平衡是否有所改善。群集亦不会删除任何多余的数据桶副本。

查看数据重新平衡的状态

如需查看数据重新平衡是否在运行，请在管理器节点上运行以下 CLI 命令：

```
splunk rebalance cluster-data -action status
```

您也可以使用管理器节点仪表板查看重新平衡状态。请参阅“使用管理器节点仪表板启动和配置重新平衡”。

配置数据重新平衡的阈值

管理器节点尝试实现合理但并非完美的平衡，确保每个对等节点上的副本数量落入一个狭小的范围内，该范围略高于或略低于所有对等节点上平均副本数量。

您可以通过 rebalance_threshold 属性（位于管理器节点的 server.conf 中）配置平衡。您可以直接在 server.conf 中或使用 CLI 命令调整设置。例如：

```
splunk edit cluster-config -mode manager -rebalance_threshold 0.95 -auth admin:your_password
```

您也可以通过管理器节点仪表板配置重新平衡阈值。请参阅“使用管理器节点仪表板启动和配置重新平衡”。

rebalance_threshold 值为 1.00 表示重新平衡将继续执行直到群集全部实现平衡，即每个对等节点上的副本数量相同。默认阈值为 0.90，表示重新平衡将继续执行直到所有对等节点都实现 90% 的完美平衡。

在 0.90 的默认设置下，重新平衡将继续执行直到所有对等节点上的平均副本数量落入 0.90 至 1.10 的范围内。例如，若您有三个对等节点，共持有 300 个副本，这表示每个对等节点平均有 100 个副本，当每个对等节点上的副本数量达到 90 至 110 个之间时，重新平衡进程就会停止。

不过，如果您倾向于 95% 的平衡，您可以把 rebalance_threshold 设置为 0.95。设置好后，管理器节点会执行必要的重新平衡，直到所有对等节点上的平均副本数量落入 0.95 至 1.05 的范围内。

群集会针对阈值，单独考量每个索引。换句话说，重新平衡的目标是确保每个索引上的平衡都达到了 rebalance_threshold 属性设置的容错。

使用管理器节点仪表板启动和配置重新平衡

您也可以通过管理器节点仪表板启动和配置重新平衡。请参阅“使用仪表板配置管理器节点”。

1. 在仪表板的右上角单击**编辑**按钮。
2. 选择**数据重新平衡**选项。
   弹出的窗口中将出现多个字段。
3. 填写必要的字段：
   - **阈值。**更改重新平衡阈值。
   - **最大运行时间。**在设置的时间段后停止重新平衡进程。如果您将该字段保留为空白，重新平衡进程将在所有对等节点都达到阈值限制后才停止。
   - **索引。**在单个索引或全部索引上运行重新平衡。
   - **可搜索。**启用搜索安全数据重新平衡。在可搜索模式中完成数据重新平衡可能需要更长时间。
4. 如需启动重新平衡，请单击**启动**按钮。

窗口中还提供重新平衡的状态信息。

控制重新平衡负载

您可以配置重新平衡的负载，把后续修复活动对对等节点索引和搜索性能的影响降到最低。

针对任何修复活动确定对等节点负载的相同属性影响能够每次重新平衡的数据桶的最大数量：max_peer_rep_load 和 max_peer_build_load 位于 [clustering] 段落中，该段落属于 server.conf。

如果属性值大于 1，数据重新平衡将使用这些属性值减 1 之后的结果。例如，若您将 max_peer_rep_load 设置为 4，则对等节点可作为一个目标最多同时参与三个重新平衡复制（而不是四个）。

冲突操作

您不能同时运行某些操作：

- 数据重新平衡
- 删除多余的数据桶
- 滚动重新启动
- 滚动升级

如果您触发其中一项操作，同时另一项操作也正在运行，那么 splunkd.log、CLI 和 Splunk Web 都会遇到问题，错误讯息指示当前存在冲突操作。

从索引器群集中删除多余的数据桶副本

多余的数据桶副本是超过群集复制因子或搜索因子的副本。例如，如果群集的复制因子为 3，则每个数据桶最好只有三个副本

在对等节点集中驻留。如果一个数据桶有四个副本，则该数据桶有一个多余的副本。

多余的副本不会干扰群集的操作，但它们是不必要的，并且需要额外的磁盘空间。

您可以通过管理器节点仪表板或 CLI 查看和删除过多数据桶副本。

警告：在删除多余的数据桶之前，请确保群集处于**完成**状态。应该有少量（如果有的话）未完成的数据桶修复任务。如果有大量多余的数据桶（数百万级），最佳做法是将群集置于维护模式，而且一次删除一个索引的多余数据桶。

多余的副本如何产生

多余的副本可能来自离开群集然后又返回群集的对等节点。当对等节点关闭时，群集会启动**数据桶修复**活动来补偿对等节点上的所有副本，因为这些副本不再对群集可用。数据桶修复的目标是将群集返回到**完整**状态，其中每个数据桶都有复制因子数目的副本和搜索因子数目的可搜索副本。

如果对等节点随后返回索引器群集，对等节点在关闭时保留的所有数据桶副本将再次对群集可用。这会导致群集保留一些数据桶的多余副本，详参“对等节点重新联机时的情况”中的介绍。

实际上，返回的对等节点会导致群集存储较执行复制因子（以及可能的搜索因子）所需更多的数据桶副本。这有时对于保持准确副本非常有用，正如该主题所解释那样，但是您可删除它们以节省磁盘空间。

使用管理器节点仪表板

查看或删除过多数据桶副本：

1. 在管理器节点上，单击 Splunk Web 右上角的**设置**。

2. 在**分布式环境**组中，单击**索引器群集化**。

这会打开管理器节点仪表板。

3. 选择“索引”选项卡。

4. 单击**数据桶状态**按钮。

这会打开数据桶状态仪表板。

5. 选择过多数据桶的索引选项卡。

该选项卡提供了过多的数据桶副本的索引列表。它枚举了有过多副本和过多可搜索副本的数据桶。它也枚举了每个类别中的总的过多副本。例如，如果索引 "new" 有一个带有三个过多副本的数据桶，其中一个副本是可搜索的，和一个带有一个过多副本的第二数据桶，其副本是不可搜索的，"new" 所在的那行就会显示：

- 带有过多副本的数据桶 2 个
- 带有过多可搜索副本的数据桶 1 个
- 过多副本共 4 个
- 过多可搜索副本共 1 个

要想从单个索引中删除过多副本，单击索引所在行的右侧的**删除**按钮。

要想从所有索引中删除过多副本，单击**删除所有过多数据桶**按钮。

使用 CLI

Splunk CLI 有两个命令帮助管理和删除过多的数据桶副本。您可以跨整个索引集合或仅仅单个索引运行这些命令。

确定群集是否有过多副本

要找出拥有过多副本的数据桶数量，包括额外的可搜索副本，从管理器节点运行以下命令：

```
splunk list excess-buckets [index-name]
```

splunk list excess-buckets 的输出类似如下所示：

```
index=_audit
       Total number of buckets=4
       Number of buckets with excess replication copies=0
       Number of buckets with excess searchable copies=0
```

```
       Total number of excess replication copies across all buckets=0
       Total number of excess searchable copies across all buckets=0
index=_internal
       Total number of buckets=4
       Number of buckets with excess replication copies=0
       Number of buckets with excess searchable copies=0
       Total number of excess replication copies across all buckets=0
       Total number of excess searchable copies across all buckets=0
index=main
       Total number of buckets=5
       Number of buckets with excess replication copies=5
       Number of buckets with excess searchable copies=5
       Total number of excess replication copies across all buckets=10
       Total number of excess searchable copies across all buckets=5
```

删除额外数据桶副本

要从群集（或群集上的一个索引）删除所有额外数据桶副本，从管理器节点运行以下命令：

```
splunk remove excess-buckets [index-name]
```

管理器节点决定从哪个节点删除额外副本。这是不可配置的，且额外副本没有必要从最近返回群集的节点删除。

冲突操作

您不能同时运行某些操作：

- 数据重新平衡
- 删除多余的数据桶
- 滚动重新启动
- 滚动升级

如果您触发其中一项操作，同时另一项操作也正在运行，那么 `splunkd.log`、CLI 和 Splunk Web 都会遇到问题，大意是当前存在冲突操作。

将对等节点置为滞留

当对等节点处于滞留状态时，其功能会减少。它停止从其他对等节点复制数据，并根据滞留类型停止对大部分或全部数据进行索引。对等节点将继续参与搜索。

对等节点可以自动进入滞留以响应可用磁盘空间不足的情况，或者手动进入滞留。

自动滞留

当对等节点自动进入滞留状态时，它

- 停止索引所有数据，包括内部的和外部的。
- 停止从其他对等节点复制数据。
- 停止参与搜索。

对等节点会在磁盘空间不足时自动进入滞留状态。控制自动滞留的设置是 `minFreeSpace`（在 `server.conf` 中）。默认值是 5000 或 5GB，这意味着当可用磁盘空间少于 5GB 时对等节点会进入滞留。

当可用磁盘空间增长到超过 `minFreeSpace` 时，对等节点自动离开滞留状态。

手动滞留

当对等节点手动进入滞留状态时，它

- 停止从其他对等节点复制数据。
- （可选）停止从使用外部数据的端口接受数据，从而导致对等节点不再索引大多数类型的外部数据。
- 继续索引内部数据并将数据流送到目标对等节点。
- 继续参与搜索。

当您手动把对等节点置为滞留状态时，它会一直处于滞留状态，直到您把它从滞留中移除。对等节点重新启动后，手动滞留仍会持续。

不接受来自外部数据端口的数据的影响

将对等节点设置为手动保留时，可以选择指示对等节点停止从使用传入数据的端口接受数据。

这会停止对大多数外部数据的索引，包括

- TCP 输入
- UDP 输入
- HEC 输入
- 数据通过**接收端口**从转发器发送到对等节点

尝试将 HEC 数据发送到对等节点会生成返回错误 "HTTPSTATUS_NOT_FOUND, 404"。

然而，外部数据可以通过以下方法继续进入对等节点：

- 脚本式输入
- 文件和文件夹监视
- receivers/stream 端点

此外，索引器可以继续将传入数据发送到另一个 Splunk Enterprise 实例或第三方系统。

使用案例

以下是手动滞留的一些主要使用案例：

- 例如，如果对等节点快耗尽空间时，要使对等节点上磁盘使用量的增长近乎停止。
- 要部分取消一个旧对等节点的配置，使其仅用于现有数据的搜索。
- 要阻止一个有麻烦的对等节点处理外部或复制的数据，同时保持对等节点可用于诊断。
- 向群集添加新的对等节点时，要强制新数据转到新的对等节点。

  注意：您也可以使用数据重新平衡将数据移动到新的对等节点。请参阅“重新平衡群集”。

- 要在属于预先批准的防火墙例外列表的对等节点上减缓磁盘使用的增长，并需要继续接收传入数据。对于此用例，可以配置对等节点停止复制活动，但是继续处理外部数据。

将对等节点置为手动滞留

要将对等节点置为滞留，请使用 CLI 命令 splunk edit cluster-config 和 -manual_detention 参数。

您可以将-manual_detention 参数设为以下几个值之一：

- on. 对等节点进入滞留，并停止从使用传入数据的端口接受数据。这些端口是接收 TCP、UDP 和 HTTP 事件收集器端口。此操作的影响是停止为大多数外部数据建立索引。对等节点继续索引内部数据。对等节点停止从其他对等节点复制数据。
- on_ports_enabled. 对等节点进入滞留，并且端口保持开放以接收传入数据。对等节点继续索引外部和内部数据。对等节点停止从其他对等节点复制数据。
- off. 对等节点不处于滞留中。此为默认值。

您可以从对等节点自身上或从管理器节点上运行此命令。

警告：在您将对等节点置为滞留之前，它必须处于 Up 状态或者 "status"。有关如何确定对等节点状态的信息，请参阅“查看管理器节点仪表板”。

要从对等节点上运行命令：

```
splunk edit cluster-config -auth <username>:<password> -manual_detention [off|on|on_ports_enabled]
```

要从管理器节点上运行命令：

```
splunk edit cluster-config -auth <username>:<password> -peers <peer_guid1>,<peer_guid2>,... -manual_detention
[off|on|on_ports_enabled]
```

请注意以下事项：

- -peers 指定要置为滞留的一组对等节点。通过对等节点的 GUID 标识每个对等节点。当您从管理器节点运行命令时，必须包含此参数。

将对等节点从手动滞留移出

要将对等节点移出滞留：

```
splunk edit cluster-config -auth <username>:<password> -manual_detention off
```

使用 REST 端点将对等节点置为手动滞留

您可以使用 REST 端点 `cluster/peer/control/control/set_manual_detention` 将对等节点置为手动滞留。

注意：已弃用之前的端点 `cluster/peer/control/control/set_detention_override`。使用在同一位置的 `cluster/peer/control/control/set_manual_detention`。

有关 `cluster/peer/control/control/set_manual_detention`，请参阅 REST API 文档。

查看滞留状态

您可以从管理器节点仪表板上查看所有对等节点的状态（滞留相关或其他情况）。请参阅“查看管理器节点仪表板”。

以下是可能的滞留状态：

- **AutomaticDetention**. 对等节点自动进入滞留。
- **ManualDetention**. 对等节点手动进入滞留，并不再处理外部数据。
- **ManualDetention-PortsEnabled**. 对等节点手动进入滞留，并继续处理外部数据。

您也可以使用 DMC 来查看对等节点的状态。

此外，一些 CLI 命令也提供对等节点的状态信息：

- 要查看所有对等节点的状态，可在管理器节点上运行此命令：

  ```
  splunk list cluster-peers
  ```

- 要查看单个对等节点的状态，可在对等节点上运行此命令：

  ```
  splunk list cluster-config
  ```

从管理器节点列表中删除对等节点

在对等节点关闭后，它仍然在管理器节点的对等节点列表中。这样的主要影响是，尽管对等节点的状态已更改为 "Down" 或 "GracefulShutdown"（具体取决于关闭的方式），但它仍会出现在管理器节点仪表板上。

您可以使用 `splunk remove cluster-peers` 命令从列表中删除对等节点：

```
splunk remove cluster-peers -peers <guid>,<guid>,<guid>,...
```

请注意以下事项：

- 所有被删除的对等节点必须处于 "Down" 或 "GracefulShutdown" 状态。
- 您可以通过逗号分隔的 GUID 列表来指定对等节点，每个对等节点指定一个。
- 可以指定 GUID 是否带连字符。例如：4EB4D230-CB8B-4DEB-AD68-CF9209A6868A 和 4EB4D230CB8B4DEBAD68CF9209A6868A 都有效。
- 如果列表中有一个无效 GUID，因为有一个 GUID 没有关联到一个停机节点，管理器节点将终止整个操作。
- 您可以通过在管理器节点上运行命令 `splunk list cluster-peers` 来获取对等节点的 GUID。

您也可以通过重启管理器节点从管理器节点列表中删除对等节点。

有关管理器节点仪表板的对等节点列表的信息，请参阅“查看管理器节点仪表板”。

管理多站点索引器群集

处理管理器站点故障

如果发生故障的站点是管理器节点所在站点，则您会丢失管理器节点的功能。您必须立即启动剩下的某个站点上的新的管理器节点。

在新的管理器节点启动之前，群集会继续以最佳状态运行。对等节点继续根据目标对等节点列表流送数据给其他节点，这些节点在管理器节点故障时仍在使用。如果一些目标对等节点有故障（和站点故障的情况类似），它们会从流出目标列表中删除故障节点，并继续流出数据给所有列表中剩下的节点。

要处理管理器站点故障，请执行以下操作：

1. 在至少一个非当前管理器节点所在站点上配置一个备用管理器节点。请参阅“在索引器群集中替换管理器节点”。**这是一个预备步骤。在需求出现之前，您必须要做这一步。**

2. 当主站点关闭时，从剩余站点中选择一个启动上面的备用管理器节点。请参阅“在索引器群集中替换管理器节点”。

3. 遵照“管理器节点重新启动或站点故障之后在多站点群集中重新启动建立索引”中的说明在群集上重新启动索引建立进程。

新的管理器节点现已完全替代了旧的管理器节点。

注意：如果发生故障的站点之后恢复了，您需要将该站点上的节点指向新的管理器节点。请参阅“确保对等节点和搜索头节点能找到新的管理器节点”。

管理器节点重新启动或站点故障之后在多站点群集中重新启动建立索引

当管理器节点重新启动时，它阻止了索引的建立，直到索引器群集上有足够多的节点满足复制因子。在一个基本的、单个站点群集中，这通常为预期行为。然而，在多站点群集的情况下，您可能想要重新启动建立索引，即便还没有足够多可用的对等节点去满足站点各方面的复制因子（例如，在站点故障的情况下）。

这种需求通常发生在这两种情况中：

- 一个站点故障，之后因为某种原因需要重新启动管理器节点。
- 管理器节点所在的站点出现故障，您在另一个站点启动备用管理器节点。

如果一个站点故障，但是运行在另一个站点的管理器节点仍在工作，建立索引照常继续，因为管理器节点只在启动时运行检查。

当与复制因子相同数量的对等节点不可用时，在管理器节点上运行 `splunk set indexing-ready` 命令以开启建立索引：

```
splunk set indexing-ready -auth admin:your_password
```

例如，假定您有一个三站点群集，配置为 `"site_replication_factor = origin:1, site1:2, site2:2, site3:2, total:7"`，其中管理器节点位于 `site1`。如果 `site2` 故障，您随后重新启动管理器节点，管理器节点在重新启动后阻止建立索引，因为它正在等待侦听 `site2` 上（`"site2:2"`）两个节点中的最小值。在这种情况下，可使用命令去重新启动剩下的站点上的索引建立。

类似的，如果有管理器节点的 `site1` 故障，您在 `site2` 上启动一个备用管理器节点，则新的管理器节点最初会阻止索引建立，因为 `site1` 不可用。然后您可使用命令让新的管理器节点去重新启动建立索引。

重要提示：在所列情况下，每次重新启动管理器节点时都必须运行 `splunk set indexing-ready` 命令。此命令仅为当前的重新启动开启建立索引。

注意：尽管此命令是为站点故障设计，您也可以使用它在单个站点群集上重新启动建立索引，只要复制因子那么多数量的对等节点可用。然而，在那种情况下，通常最好是等待，直到复制数量的节点重新加入群集。

将多站点索引器群集转换为单个站点群集

您可以将一个多站点索引器群集转换成一个基本的、单个站点群集。当您这样做时，所有节点和搜索头都变成非显式单个站点的一部分。

1. 停止所有群集节点（管理器节点/对等节点/搜索头）。

2. 在管理器节点上，编辑 `server.conf`：

a. 将 multisite 设置为 false。

b. 设置单个站点的 replication_factor 和 search_factor 属性以实现所需的复制行为。

c. 删除 site 属性。

3. 在每个搜索头上，编辑 server.conf：

a. 将 multisite 设置为 false。

b. 删除 site 属性。

4. 在每个对等节点上，编辑 server.conf：

a. 删除 site 属性。

5. 启动管理器节点。

6. 启动对等节点和搜索头。

请注意以下事项：

- 管理器节点忽略留在 server.conf 中的任何多站点属性（site_replication_factor 等等）。
- 重新启动后，管理器节点会为每个数据桶选取一个主要副本。
- 转换后，任何超出单个站点复制因子的数据桶副本留在原处。有关删除这些额外副本的信息，请参阅“删除过多的数据桶副本”。
- 将来的数据桶复制和数据桶修复将遵循为单个站点复制和搜索因子设定的值。

关于如何将单个站点群集转换为多站点的信息，请参阅“将索引器群集从单个站点迁移到多站点”。

将对等节点移到新站点

使用此过程将对等节点重新定位到另一个站点。如果节点被发往一个错误站点，这将非常有用。且只有在节点被部署到群集上以后才能发现错误。

1. 使用 offline 命令使对等节点脱机，详参“使对等节点脱机”中的介绍。管理器节点将重新分配被此节点掌管的数据桶副本，将其分给同一站点的其他节点。

2. 将节点服务器运送到新站点。

3. 从服务器删除整个 Splunk Enterprise 安装，包括其带有所有数据桶副本的索引数据库。

4. 在服务器上重新安装 Splunk Enterprise，重新启用群集化，并将节点的站点值设置为新站点位置。

对等节点作为一个新的节点重新加入群集。

在多站点索引器群集中取消站点配置

要取消站点配置，您可重新配置若干个站点特定的属性。

取消站点配置

警告：在继续之前，请注意下列问题：

- 如果已取消配置的站点上有某个对等节点包含任何在该节点加入群集前就已创建的数据桶，则这些数据桶仅存在于该对等节点上；站点取消配置后，这些数据桶将丢失。
- 类似地，若已取消配置的站点在加入多站点群集前为单个站点群集，则该站点为单个站点群集期间创建的任何数据桶仅存在于该站点上；站点取消配置后，这些数据桶将丢失。
- 若在取消配置进程临近结束时重新启动管理器节点，管理器节点将开始数据桶修复活动，由此把群集返回到**完整**状态。该操作将耗费相当长时间，尤其如果已取消配置的站点持有大量原始数据桶。

前提条件

取消群集中某个站点的配置前，该群集必须满足以下条件：

- 群集必须处于完整状态。

- 管理器节点不得位于您想要取消其配置的站点上。如果管理器节点刚好位于您想要取消其配置的站点上，请遵照“处理管理器节点站点故障”中的指导说明进行操作。
- 必须配置 site_replication_factor 属性，这样才能确保每个数据桶至少有一个副本驻留在未计划取消配置的站点上。例如，在一个包含两个站点的群集中，有效的配置为 site_replication_factor = origin:1,total:2。
- 必须配置 site_search_factor 属性，这样才能确保每个数据桶至少有一个可搜索副本驻留在未计划取消配置的站点上。例如，在一个包含两个站点的群集中，有效的配置为 site_search_factor = origin:1,total:2。
- 如果您需要重新配置 site_replication_factor 或 site_search_factor 以便所有数据桶在其他站点上都有副本，您必须等到管理器节点完成修复活动并把群集返回至完整状态，才能继续取消配置的操作。

步骤

1. 对于站点中的每个搜索头，请禁用该搜索头或更改搜索头 site 属性，由此来指定一个剩余站点。例如，把搜索头的站点更改为 site2：

   ```
   splunk edit cluster-master https://10.160.31.200:8089  -site site2
   ```

   请参阅“配置搜索头”。

2. 对于使用索引器发现并指定已取消配置站点的每个转发器，请更改其 site 属性，由此来指定一个剩余站点。例如，把转发器的站点更改为 site2：

   ```
   [general]
   site = site2
   ```

   您必须重新启动转发器，所做的配置更改才会生效。请参阅“在多站点群集中使用索引器发现”。

3. 在管理器上运行 splunk enable maintenance-mode。本步骤可以防止不必要的数据桶修复。请参阅“使用维护模式”。
4. 要确认管理器节点进入了维护模式，运行 splunk show maintenance-mode。
5. 在管理器节点上更新下列属性：
   - available_sites
   - site_replication_factor
   - site_search_factor
   - site_mappings

   有关必要更新的详细信息，请参阅“重新配置属性”。
6. 重新启动管理器。重启之后，所做的属性更改将生效。
   注意：管理器节点重新启动后，已取消配置站点上的对等节点若尝试重新加入群集，会以失败告终。请忽略结果消息。
7. 在管理器上运行 splunk disable maintenance-mode。此步骤为剩余站点上的对等节点启动修复活动。
8. 要确认管理器节点退出了维护模式，运行 splunk show maintenance-mode。
9. 在已取消配置站点的所有对等节点上运行 splunk stop。您可立即将这些对等节点从站点中移除。
10. 如需验证取消配置是否成功，请查看管理器节点仪表板的顶部。上面应该会说明群集已满足搜索因子和复制因子。两个因子都满足后，群集会处于完整状态，说明取消配置操作成功。请参阅“查看管理器节点仪表板”。
    注意：因为站点取消配置通常涉及大量数据桶修复活动，所以群集需要相当长的时间才能返回至完整状态。

重新配置属性

取消一个站点的配置时，您必须更改管理器节点上 server.conf 内的多个站点特定属性：

- available_sites：从该属性的站点列表中删除取消了配置的站点。
- site_replication_factor 和 site_search_factor：这两个属性中如果有显式站点取消了配置，请删除该站点并在必要时重新配置属性。
- site_mappings：把已取消了配置的站点的映射添加到此属性。请参阅“映射取消了配置的站点”。

更改上述任意属性后，您必须重新启动管理器节点。

映射取消了配置的站点

取消了一个站点的配置后，此站点的原始数据桶副本仍绑定至该已取消了配置的站点，除非您将此站点映射到剩余的一个活跃站点。这样做会使得群集无法满足其复制因子或搜索因子。

要解决这个问题，您可以把已取消配置的站点映射到活跃的站点。映射之后，源自该已取消配置的站点的数据桶副本将被复制到映射指定的活跃站点，让群集得以再次满足其复制因子和搜索因子。

注意：site_replication_factor 属性和 site_search_factor 属性决定群集的原始数据桶副本数量。

语法

在您为站点取消配置之前，请先映射此站点。请参阅“取消站点配置”。

要将一个计划取消配置的站点映射到一个剩余站点，可使用 server.conf 中的 site_mappings 属性。只能在管理器节点上设置该

属性，语法如下：

```
site_mappings = <comma-separated string>
```

请注意以下事项：

- <comma-separated string> 包含从已取消配置的站点到剩余活跃站点的映射。这些映射可为以下两种类型中的一种：
  - <decommissioned_site_id>:<active_site_id>. 例如，site2:site3，其中 site2 为已取消配置的站点，site3 为活跃的站点。此类映射称之为显式映射。显式映射可以有很多种。
  - default_mapping:<active_site_id>. 例如，default_mapping:site4，其中 site4 为活跃的站点。最多只有一个默认的映射。建议您始终包含一个默认的映射，将其用作错误或缺失的显式映射的回退。
- <decommissioned_site_id>:<active_site_id> 的情况下，<decommissioned_site_id> 内的原始数据桶副本将从剩余站点被复制到 <active_site_id> 上的对等节点。此操作可让群集满足其复制因子和搜索因子的要求。
- default_mapping:<active_site_id> 的情况下，若已取消配置的站点上没有显式映射，则该站点的原始数据桶副本将被复制到 <active_site_id>。
- 如果映射中的一个活跃站点稍后取消了配置，则该站点的旧映射必须重新映射到当前处于活跃状态的站点。例如，在 site2:site3 的情况下，如果 site3 本身已取消了配置，您必须把旧映射 site2:site3 替换为一组新的映射；新的映射使用字符串 site2:site4,site3:site4 把 site2 和 site3 都映射到一个活跃站点，如 site4。

更改此属性后请重新启动管理器节点。

示例

以下示例假设群集原本有五个站点，从 site1 到 site5。

- **"site_mappings = site2:site3"** 此配置把已取消配置 site2 映射到活跃站点 site3。此映射操作会把 site2 的原始数据桶副本复制到 site3。没有默认的站点映射。
- **"site_mappings = site1:site3,default_mapping:site4"** 此配置把已取消配置的 site1 映射到 site3，并把其他所有已取消配置的站点映射到 site4。此映射操作会把各已取消配置的站点的原始数据桶副本复制到它们各自的映射站点。
- **"site_mappings = default_mapping:site5"** 此配置把所有已取消配置的站点映射到 site5。此映射操作会把所有已取消配置的站点的原始数据桶副本复制到 site5。

索引器群集如何工作

高级用户的基本索引器群集概念

要了解群集的功能，需要熟悉几个概念：

- **复制因子。**它指定群集保留的数据副本数量。它影响着群集的复原设置，即群集承受多种节点故障的能力。
- **搜索因子。**它指定**可搜索**数据副本的数量。它影响着群集从故障节点中恢复的速度。
- **数据桶。**这些是索引的基本存储容器。它们与索引器数据库中的子目录相对应。
- **群集状态。**这些状态说明了群集的运行状况。

您可以在有关群集架构的介绍主题“基本索引器群集架构”中找到这些概念的概述。您现在正在阅读的章节中的主题提供了更为详细的信息。

复制因子

在设置索引器群集时，指定群集要保留的数据副本数量。对等节点将传入数据存储在**数据桶中**，群集保留每个数据桶的多个副本。群集将每个数据桶副本存储在一个单独的对等节点上。群集维护的每个数据桶的副本的数量就是**复制因子**。

复制因子和群集复原

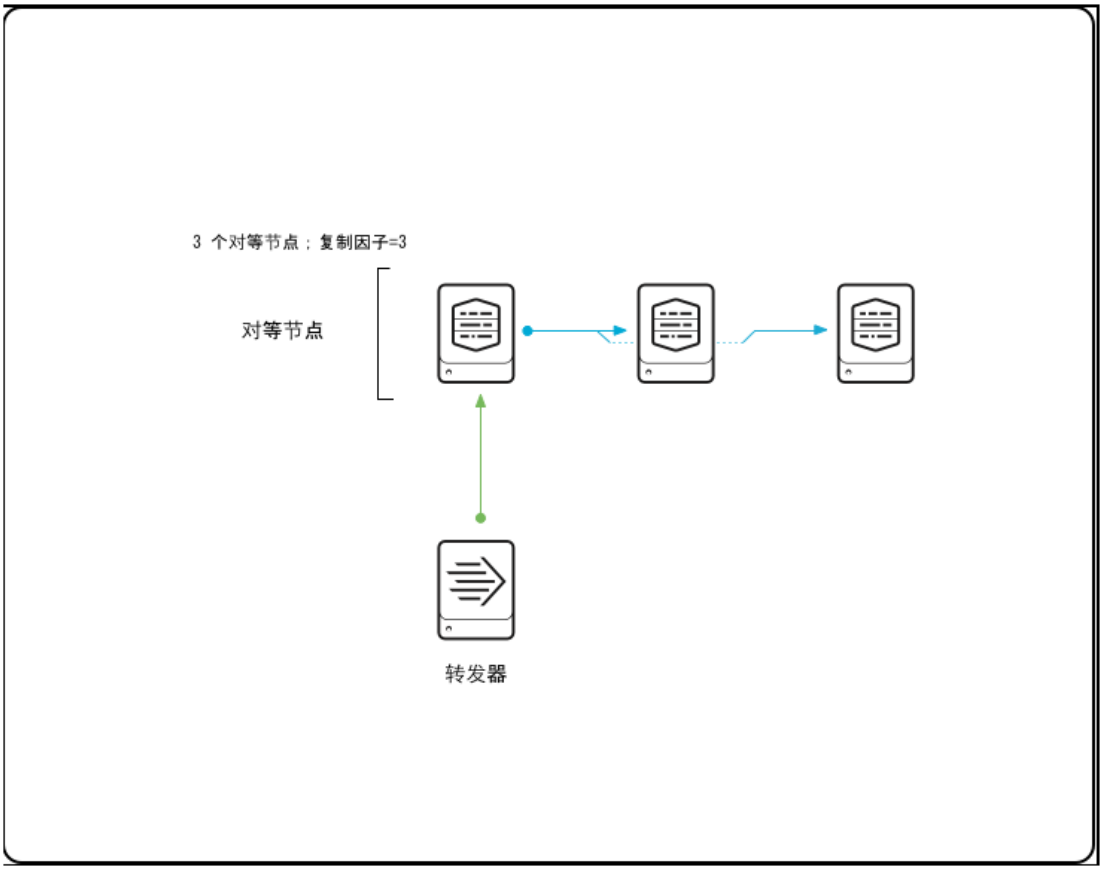
群集可以容许（复制因子 – 1）个对等节点出现故障。例如，要确保系统可以容许两个对等节点出现故障，必须将复制因子配置为 3，这意味着群集将每个数据桶的三个完全相同的副本存储单独的节点上。复制因子为 3 时，您可以肯定的是如果群集中发生故障的对等节点不超过两个，所有数据都将可用。当两个节点都关闭时，在剩余对等节点上仍有一个完整的数据副本可用。

增加复制因子可以增加系统容许的对等节点故障数量。复制因子为 2 时，只能容许一个节点故障；复制因子为 3 时，可以容许两个并发故障，依此类推。

问题是您需要存储和处理所有数据副本。尽管复制活动不会消耗太多的处理能力，但是随着复制因子的增加，您仍需要运行更多的索引器并为索引数据置备更多的存储。另一方面，由于数据复制本身需要的处理能力较少，因此您可以利用群集中的多个索引器来获取和索引更多的数据。群集中的每个索引器既可以充当来源索引器（“数据来源对等节点”），又可以充当复制目标（“目标对等节点”）。它不但可以为传入数据建立索引，还可以存储来自群集中其他索引器的数据副本。

示例：操作中的复制因子

在下图中，一个对等节点在接收来自转发器的数据，它对此数据进行处理，然后流送到其他两个对等节点。群集中将包含对等节点数据的完整副本，每个对等节点上一个副本。



注意：该图以高度简化的方式表示对等节点复制，其中所有数据将通过单个对等节点进入系统中。有几个问题会添加现实应用的复杂性：

- 在大多数群集中，每个对等节点都可充当数据来源对等节点和目标对等节点，既从转发器接收外部数据，又从其他对等节点接收复制的数据。
- 为进行横向扩展，复制因子为 3 的群集可能包含三个以上的对等节点。在任意指定时间，每个数据来源对等节点使其数据的副本流入两个目标对等节点，但每次启动一个新的热数据桶时，它的目标对等节点集都可能发生变化。

本章后面的几个主题详细说明群集如何处理数据。

多站点群集中的复制因子

多站点群集使用复制因子的特殊版本：站点复制因子。站点复制因子不仅决定整个群集维护的副本数量，而且还决定每个站点维护的副本数量。关于站点复制因子的信息，请参阅“配置站点复制因子”。

搜索因子

配置管理器节点时，指定**搜索因子**。搜索因子决定了索引器群集保留的**可搜索**的数据副本的数量。也就是说，搜索因子决定每个**数据桶**的可搜索副本的数量。搜索因子的默认值为 2，这意味着群集会保留所有数据的两个可搜索副本。搜索因子必须小于或等于**复制因子**。

可搜索和不可搜索数据桶副本

数据桶的可搜索副本与**不可搜索**副本之间的差异如下：可搜索副本包含数据本身，以及对等节点用来搜索数据的一些广泛的索引文件。不可搜索副本只包含数据。即使数据存储在不可搜索副本中，但是已经执行了初步处理并采用了适当存储形式，以便在以后需要时可以创建索引文件。有关组成 Splunk Enterprise 索引的文件的更多信息，请阅读子标题“数据文件”。

从对等节点故障中恢复搜索

搜索因子至少为 2 时，如果某个对等节点故障，群集可以在几乎没有中断的情况下继续搜索。例如，指定复制因子为 3，搜索因子为 2。群集将维护群集中单独的对等节点上的所有数据桶的三份副本，每个数据桶的两份副本是可搜索副本。然后，如果一个对等节点故障，并且其中包含已参与搜索的数据桶副本，该数据桶在另一个对等节点上的可搜索副本可立即加入和开始参加搜索。

另一方面，如果群集的搜索因子仅为 1，并且某个对等节点故障，则在对完整的群集数据集合恢复搜索之前将存在明显的滞

后。虽然可以将数据桶的不可搜索副本变为可搜索副本，但这样做会花费时间，因为必须首先从原始数据文件建立索引文件。如果出现故障的对等节点存储了大量的可搜索数据，处理时间可能很长。有关评估使不可搜索副本成为可搜索副本所需时间的帮助，请查看此处。

您可能想要限制群集上可搜索副本的数量，原因是与不可搜索数据相比，可搜索数据将占用更多存储空间。因此，您需要在快速访问所有数据以防出现对等节点故障与增加的存储需求之间权衡。有关可搜索和不可搜索数据的相对存储大小的帮助，请阅读“存储注意事项”。对于大多数需求来说，搜索因子的默认值 2 是比较合适的权衡。

多站点群集中的搜索因子

多站点群集使用搜索因子的特殊版本：站点搜索因子。站点复制因子不仅决定整个群集维护的可搜索副本数量，而且还决定每个站点维护的可搜索副本数量。关于站点搜索因子的信息，请参阅“配置站点搜索因子”。

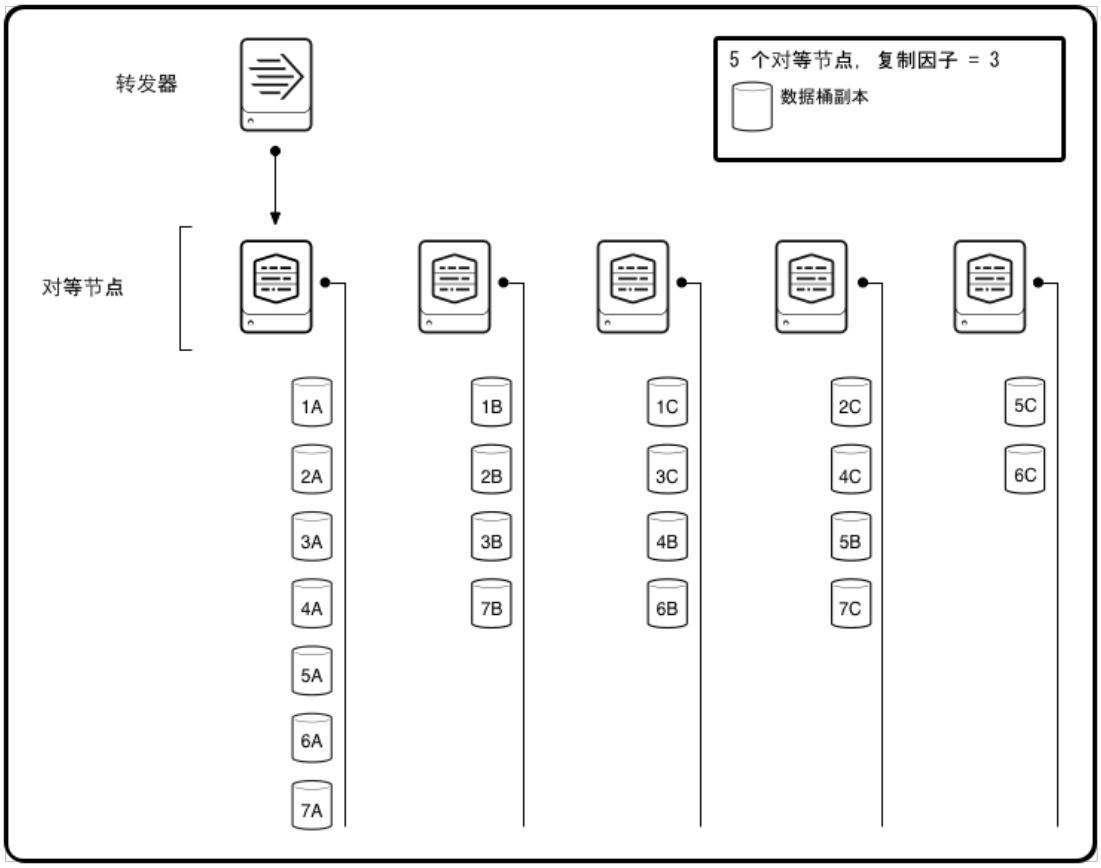
数据桶和索引器群集

Splunk Enterprise 将索引数据存储在**数据桶**中，数据桶是指包含数据以及数据索引文件的目录。索引通常包含许多数据桶，按数据的时间组织。

索引器群集按数据桶复制数据。原始数据桶副本及其在其他对等节点上的复制副本包含相同的数据组，但只有**可搜索**副本另外还包含索引文件。

在群集中，来自单个数据来源对等节点的数据桶副本可以分散在许多目标对等节点中。例如，如果群集中有五个对等节点且复制因子为 3（横向扩展的典型情况），群集将为每个数据桶保留三个副本（数据来源对等节点上的副本和两个目标对等节点上的复制副本）。每次数据来源对等节点启动新的热数据桶时，管理器节点都会为对等节点提供一组新的目标对等节点用来将数据复制到其中。因此，当原始副本都在数据来源对等节点上，那些数据桶的复制的副本将被随机分布到其他节点。此行为是不可配置的。可以确定的一点是，相同对等节点上绝不会存在相同数据桶的两个副本。对于多站点群集的情况，也可配置复制的副本的站点位置，但是仍然不能指定实际节点的位置。

下图显示了刚刚描述的方案-五个对等节点、复制因子为 3、七个原始的数据来源数据桶，且原始数据桶的副本分散在所有对等节点中。为减少复杂性，图中仅显示了来自一个对等节点的数据所对应的数据桶。在现实方案中，其他大多数的对等节点（如果不是所有）也会成为原始数据并复制到群集上的其他对等节点。



在此图中，1A 是数据来源数据桶。1B 和 1C 是该数据桶的副本。图中对 2A/B/C、3A/B/C 等使用相同的约定。

您只有深刻了解数据桶才能了解群集架构。本部分的其余内容介绍对于群集部署特别重要的一些数据桶概念。有关数据桶的全面介绍，请阅读“索引器如何存储索引”。

数据文件

数据桶中的文件有以下两种重要类型：

- 压缩形式的已处理外部数据（**原始数据**）
- 指向原始数据的索引（**索引文件**，也称为 **tsidx 文件**）

数据桶也包含一些其他类型的文件，但是这些文件是需要理解的最重要的类型。

原始数据实际上并非如词典所定义的词语“原始”数据一样。而是指在被处理到事件中后即包含外部数据。已处理的数据存储在压缩的原始数据日志文件中。作为日志文件，除了包含**事件数据**以外，原始数据文件还包含生成相关索引文件（如果这些文件缺失）所需的全部信息。

无论**可搜索**还是**不可搜索**，所有数据桶副本都包含原始数据文件。可搜索副本还包含索引文件。

某一对等节点从转发器收到数据块后，它会处理数据，并将其添加到其本地热数据桶中的原始数据文件。该节点还会为数据建立索引，创建相关的索引文件。另外，该节点只使已处理原始数据的副本流入其每个目标对等节点，这些目标对等节点又将数据添加到自己的数据桶副本的原始数据文件中。原始数据桶副本和复制的数据桶副本中的原始数据是相同的。

如果群集的搜索因子为 1，则目标对等节点只在数据桶副本中存储原始数据。它们不为数据生成索引文件。通过不将索引文件存储在目标对等节点上，可以限制存储要求。由于原始数据以日志文件形式进行存储，因此如果保留完全索引原始数据的对等节点故障，一个目标对等节点会介入，并从其原始数据副本生成索引。

如果群集的搜索因子大于 1，则部分或全部的目标对等节点还会为数据创建索引文件。例如，假如有复制因子 3，搜索因子 2。在这种情况下，数据来源对等节点将它的原始数据流式发送到两个目标对等节点。然后，其中的一个对等节点将使用原始数据来创建索引文件，该对等节点将这些索引文件存储在其数据桶的副本中。这样，将有两个可搜索数据副本（原始副本和带有索引文件的复制的副本）。如“搜索因子”所述，这样在出现对等节点故障的情况下允许群集更迅速地恢复。有关可搜索数据桶副本的更多信息，请参阅本主题后面的“数据桶可搜索性”。

有关数据桶文件的更多信息，请参阅这些主题：

- 有关在对等节点关闭时如何生成数据桶文件的信息，请阅读“对等节点故障时的情况”。
- 有关原始数据和索引文件相对大小的信息，请参阅存储注意事项。

数据桶阶段

数据桶老化时会经历多个阶段：

- 热
- 温
- 冷
- 冻结

有关这些阶段的详细信息，请阅读“索引器如何存储索引”。

对于即将讨论的群集架构，您只需基本了解这些数据桶阶段即可。热数据桶是仍然在向其写入信息的数据桶。索引器完成写入热数据桶（例如，因为数据桶达到最大大小）时，会将数据桶滚动到温，并开始向新的热数据桶写入。温数据桶是可读的（例如，用于搜索），但索引器不向其中写入新数据。最终，数据桶滚动到冷，然后是冻结，此时数据桶会被归档或删除。

请务必牢记以下几项其他详细信息：

- 热/温和冷数据桶存储在单独的可配置位置。
- 温数据桶或冷数据桶的文件名中包括数据桶中数据的时间范围。有关数据桶命名约定的详细信息，请阅读“索引目录的结构”。
- 搜索将发生在热数据桶、温数据桶和冷数据桶中。
- 导致数据桶滚动的条件可进行配置，如“配置索引存储”中所述。
- 有关存储硬件信息（例如评估存储要求的帮助）请阅读“存储注意事项”。

数据桶可搜索性和主要性状态

数据桶的副本可以是**可搜索的**，也可以是**不可搜索的**。由于群集会保留一个数据桶的多个可搜索副本，因此群集需要有办法可以识别哪个副本参与了搜索。为此，群集使用了**主要性**的概念。一份可搜索数据桶副本可以是**主要副本**，也可以是非主要副本。

如果某一数据桶副本同时包含索引文件和原始数据文件，则该数据桶副本是可搜索的。接收外部数据的对等节点会对原始数据建立索引，还会将原始数据的副本发送到其对等节点。如果搜索因子大于 1，则其中部分或所有对等节点还将为其所复制的数

据桶生成索引文件。因此，例如，如果复制因子为 3，搜索因子为 2，群集**完整**，群集包含每个数据桶的三个副本。所有三个副本都包含原始数据文件，其中两个副本（位于数据来源对等节点上的副本和目标对等节点上的一份副本）还包含索引文件，从而是可搜索副本。第三方副本是不可搜索副本，但如果需要可使其成为可搜索副本。不可搜索副本变为可搜索副本的主要原因是数据桶的可搜索副本所在的对等节点故障。

数据桶的主要副本是参与搜索的可搜索副本。单个站点**有效**群集中每个数据桶只有一个主要副本。这样，每个数据桶有且仅有一个副本会发生搜索。如果包含主要副本的节点关闭，则可以立即将其他节点上的可搜索但非主要副本指定为主要副本，从而使搜索能够继续进行而无需首先等待生成新的索引文件。

注意：对于多站点群集的情况，有效群集是指在每个支持搜索相关性的站点上拥有一组主要副本的群集。在搜索相关性中，搜索头在本地站点上的对等节点之间进行搜索。这需要每个站点有它自己的主要数据桶集合。

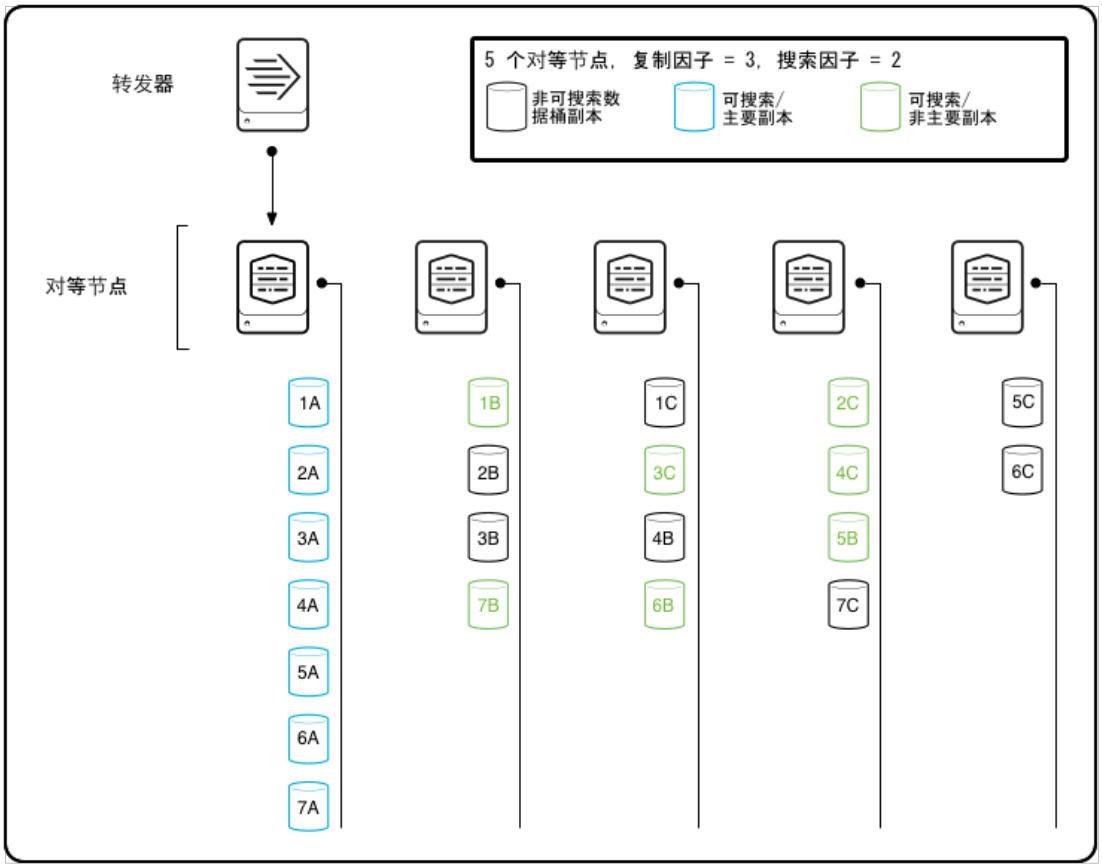
最初，数据源自的对等节点上的数据桶的副本是主要副本，但可随时间变更。例如，如果节点故障，管理器节点将主要性从已关闭的对等节点上的主要副本重新分配给剩余对等节点上的相应的可搜索副本。有关此过程的更多信息，请阅读“对等节点故障时的情况”。

当管理器节点重新平衡群集以尝试实现主要副本在一组对等节点的更加均匀的分布时，还会发生主要性重新分配。在这些情况下，将出现重新平衡：

- 对等节点加入或重新加入群集。
- 管理器节点重新加入群集。
- 您手动调用管理器节点上的 `rebalance_primaries` REST 端点。

有关详细信息，请参阅“重新平衡索引器群集的主要数据桶”。

下图显示了分散在所有对等节点中的数据桶（如上图所示）。群集的复制因子为 3 且搜索因子为 2，这意味着群集将保留每个数据桶的两个可搜索副本。此处，数据来源对等节点上的数据桶副本全部都是主要副本（因此还是可搜索的）。数据桶的另一个可搜索（但非主要）副本分散在群集中的大部分剩余对等节点中。



这组主要数据桶副本定义了群集的生成期间（如下一部分中所述）。

生成期间

生成期间确定了群集数据桶的哪些副本是主要副本，因而将参与搜索。

注意：实际发生搜索的数据桶集还取决于搜索时间范围等其他因素。这适用于任何索引器，无论是群集还是非群集。

随着对等节点离开和加入群集，生成期间也会更改。某个对等节点关闭时，它的主要数据桶副本会重新分配给其他对等节点。管理器节点也会在某些其他情况下重新分配主要副本，其过程被称为“群集重新平衡”。

以下是定义生成期间的另一种方法：生成期间是群集的**有效**状态的快照；“有效”意味着群集中的每个数据桶只有一个主要副本。

当前在管理器节点注册的所有对等节点都会参与当前的生成期间。某个对等节点加入或离开群集时，管理器节点会创建新的生成期间。

注意：由于给新的数据桶副本重新分配主要性的过程不是瞬时的，当一个事件例如对等节点停机导致重新分配主要性，尤其是在大量主要性正驻留在停机节点上的情况下，群集可能会很快经过多种生成期间。

生成期间是群集范围的属性。在一个多站点群集中，所有站点的生成期间值都是一样的。

群集节点如何使用生成期间

下面是各种群集节点如何使用生成期间信息：

- 管理器节点会创建每个新的生成期间，并为其分配**生成期间 ID**。需要时，它会将当前生成期间 ID 传递给对等节点和搜索头。它还会跟踪每个生成期间的主要数据桶副本，以及这些副本所在的对等节点。
- 对等节点会跟踪每个生成期间自己的哪些数据桶是主要数据桶副本。对等节点会保留多个生成期间的主要信息。
- 对于每个搜索，搜索头会使用从管理器节点获得的生成期间 ID 来确定要执行搜索的对等节点。

生成期间更改的情况

生成期间会在以下情况下发生变化：

- 管理器节点联机。
- 对等节点加入群集。
- 某一对等节点有意地（通过 CLI 命令 `offline`）或无意地（由于崩溃）关闭。某个对等节点故障时，管理器节点会将主要性从已故障的节点上的数据桶副本重新分配给剩余节点上的相同数据桶的可搜索副本，并创建新的生成期间。
- 在重新平衡主要副本的任何时候，如当您手动按管理器节点上的 `rebalance_primaries` REST 端点时。有关重新平衡的信息，请参阅“重新平衡索引器群集的主要数据桶。”
- 管理器节点解决了某些数据桶异常时。

只有在数据桶从热滚动到温时管理器节点才不创建新的生成期间，从而使新的热数据桶得以创建（除非出于上述列出的原因之一滚动数据桶）。在这种情况下，对等节点组不会更改。搜索头只需要了解哪些对等节点是生成期间的一部分；即哪些对等节点目前参与了群集。它不需要了解特定对等节点上的哪些数据桶副本是主要副本；对等节点自身会跟踪该信息。

如何在搜索中使用生成期间

搜索头会定期轮询管理器节点以获取最新的生成期间信息。当生成期间更改时，管理器节点会为搜索头提供新的生成期间 ID 以及属于该生成期间的对等节点的列表。每个搜索头又会在启动搜索时将此 ID 提供给对等节点。对等节点使用此 ID 来确定在该搜索中它们的哪些数据桶是主要数据桶。

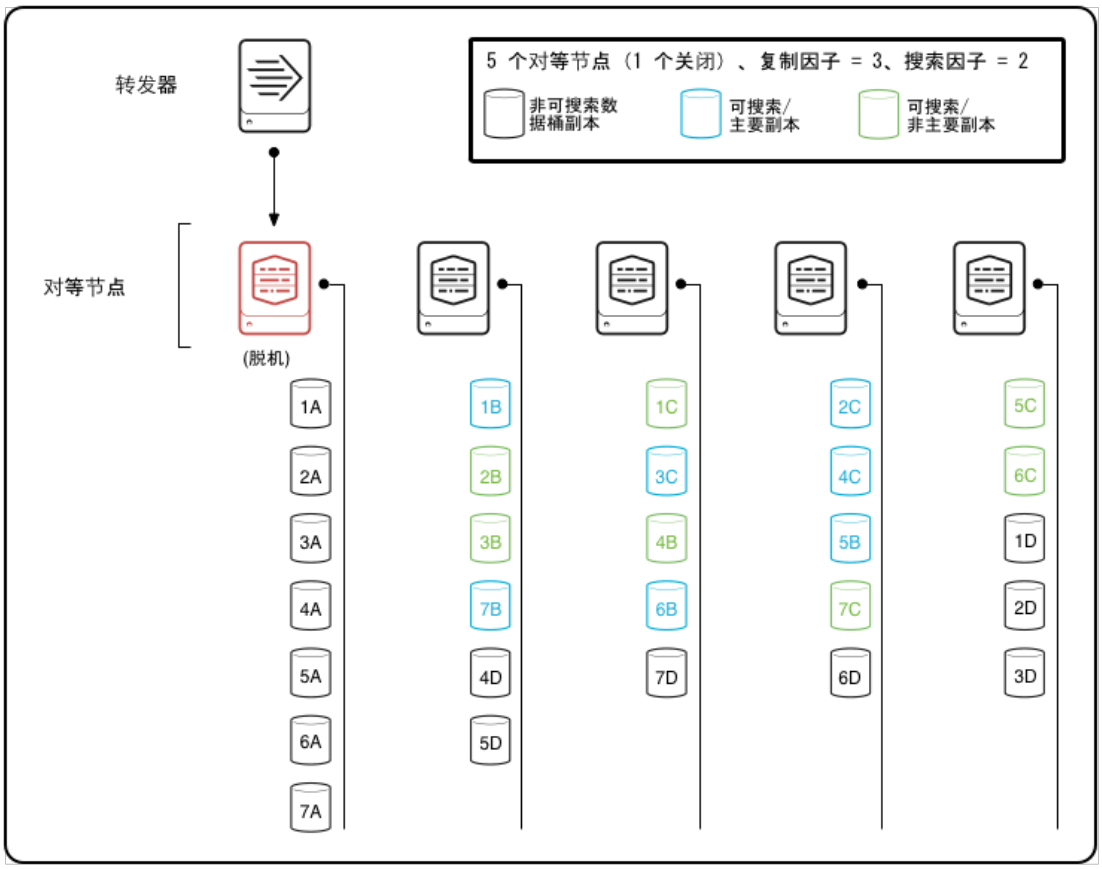
通常，搜索在主要数据桶副本的最近生成期间内发生。但是，如果是长时间运行的搜索，搜索可能在早期生成期间内运行。这种情况通常会发生，因为对等节点在搜索期间发生故障。这样，即使某些数据缺失（由于对等节点故障），长时间运行的搜索也能够完成。另一种方法是重新启动搜索，如有必要，您始终可以手动这样做。

发生故障的对等节点为什么会导致生成期间更改

发生故障的对等节点导致管理器节点创建新的生成期间的原因在于，对等节点发生故障时，管理器节点将发生故障的对等节点的主要副本重新分配给其他节点上的副本。在先前的生成期间不是主要副本的副本会在新的生成期间中成为主要副本。通过了解与搜索相关的生成期间 ID，对等节点可以确定自己的哪些数据桶是该搜索的主要数据桶。

例如，下文的图中所示为一个群集的简化版本，在所有主要副本所在的数据来源节点已发生故障，并且管理器节点已指示剩余的对等节点修复数据桶后，该群集仍与之前相同。首先，管理器节点将主要性重新分配给每个数据桶的剩余可搜索副本。接下来，它指示对等节点使其不可搜索副本成为可搜索副本，以弥补缺失的一组可搜索副本。最后，它指示对一组新的不可搜索副本（1D、2D 等）进行复制，在剩余对等节点之间分散。

即使数据来源节点发生故障，群集仍然能够完全恢复其**完整**和**有效**状态，条件是群集的数据桶副本总数（即复制因子数目）为 3，可搜索的数据桶副本数（即搜索因子数目）为 2，每个数据桶有正好一个主要副本。此图中所表示的生成期间与上图有所不同，因为主要副本已经移到了不同的对等节点。



注意：该图仅显示了来自其中一个对等节点的数据桶。此图表较为完整的版本会显示来自多个对等节点的数据桶在群集中发生迁移的情况。

群集如何处理冻结数据桶

如果是在独立索引器上，当一个数据桶滚动到冻结时，索引器会从 `colddb` 目录中删除它。根据它的退休策略，索引器在删除它之前可能会将它复制到归档目录。请参阅“归档索引的数据”。

如果是索引器群集，当对等节点冻结了一个数据桶副本时，它会通知管理器节点。管理器节点随后停止针对该数据桶的修复活动。假设的条件是其他对等节点还将最终冻结它们的该数据桶副本。如果冻结行为由 `maxTotalDataSizeMB` 属性决定，而该属性限制了索引的最大大小，那么可能需要一段时间才能完成要冻结数据桶的所有副本，因为不同对等节点上的索引大小通常不同。因此，该索引可能在某个对等节点上达到其最大的大小，导致最旧数据桶遭冻结，尽管该索引在其他对等节点上仍未超过大小限制。

注意：为了延长数据桶用于搜索的时间，6.3 版本更改了群集响应遭冻结的主要数据桶副本的方式：

- 在 6.3 版本之前的群集中，当主要副本冻结时，群集不再尝试把该主要副本重新分配给其他任意剩余的可搜索副本。一旦主要副本冻结，数据桶上的搜索活动也将停止。
- 在 6.3 和更高的版本中，当主要副本冻结时，如果存在另一个可搜索副本，群集将会把主要副本重新分配给该可搜索副本。数据桶添加新的主要副本后，上面的搜索活动将继续执行。新的主要副本亦冻结之后，群集将尝试再次把此主要副本重新分配给另一个可搜索副本。一旦数据桶的所有可搜索副本都冻结后，该数据桶上的搜索活动将停止。

在 6.3 之前和之后的版本中，群集在副本冻结后都会停止数据桶上的修复活动，亦即，群集不会再为了满足数据桶的复制因子和搜索因子而尝试创建新的副本，或把不可搜索副本转换为搜索副本。

索引器群集状态

处于良好运行顺序的索引器群集是**有效**而又**完整**的群集：

- **有效**群集中每个数据桶只有一个**主要**副本。对于多站点群集的情况，有效群集在每个支持搜索相关性的站点上拥有一组主要副本的群集。
- **完整**群集每个数据桶具有与**复制因子**相同的副本数，以及与**搜索因子**相同的**可搜索**副本数。对于多站点群集的情况，数据桶副本的数量也必须满足站点特定的复制和搜索因子的需要。

请注意这些要点：

- 有效群集能够处理整个数据集的搜索请求。有效多站点群集也要实现任何内在的搜索相关性的目标。
- 完整群集符合故障容错的指定要求。
- 完整群集还是有效群集，但是有效群集不一定完整。

此外，为确保稳健的数据可用性，群集不能只具有完整性，还必须至少将其搜索因子设置为 2。这样，在一个对等节点发生故障时，可以确保搜索头可以继续搜索整个群集，而不发生中断。

当某一对等节点故障时，管理器节点会指示群集执行用于恢复其有效和完整状态的活动。在某些情况下，群集可能可以恢复到有效状态，但无法恢复到完整状态。（例如，假设群集有三个对等节点，复制因子为 3。如果一个对等节点发生故障，只要此对等节点仍未恢复正常，该群集就无法恢复其完整状态，但可以恢复其有效状态。）有关群集如何从故障节点中恢复的详细信息，请参阅“对等节点故障时的情况”。

群集索引如何工作

在讨论数据和信息在建立索引期间如何在节点间流动时，区分对等节点扮演的两个角色是非常有用的：

- **数据来源节点。**数据来源节点从转发器或其他外部数据来源获取数据。
- **目标对等节点。**目标对等节点从数据来源节点接收复制的数据流。

实际上，单个对等节点经常同时充当数据来源节点和目标对等节点。

重要提示：在典型索引器群集部署中，所有对等节点都是数据来源节点；即，每个节点都有自己的外部输入集。这不是一项要求，但通常是最佳做法。没有理由保留某些对等节点只是为了用作目标对等节点。存储复制的数据的处理成本最小，在任何情况下，当前都无法指定哪些节点将接收复制的数据。管理器节点会按数据桶确定此信息，但此行为不可配置。您必须假定所有对等节点都将作为目标。

注意：除了复制外部数据以外，每个对等节点会以相同方式将其内部索引复制到其他对等节点。简单起见，这里只讨论外部数据。

如何选择目标对等节点

当数据来源对等节点启动一个热数据桶时，管理器节点会给它一个目标对等节点的列表，它会将复制的数据流送到这些节点。该列表的数据桶是特定的。如果数据来源对等节点正将数据写入若干个热数据桶，则每个数据桶的内容可以流送到不同的目标对等节点集。

管理器节点随机选择目标对等节点列表。如果是在多站点群集化中，它会遵守站点限制（如复制因子所指定的），但是会在这些约束条件内随机选择目标对等节点。

当对等节点启动时

当某对等节点启动时会发生以下事件：

1. 该对等节点向管理器节点注册并从管理器节点接收最新的**配置软件包**。

2. 管理器节点将重新平衡整个群集的**主要**数据桶副本并开始新生成期间。

3. 该对等节点开始插入外部数据，方式与任何索引器相同。它将数据处理成事件，然后将数据附加到原始数据文件中。此外，还会创建关联的索引文件，并将这些文件（原始数据文件和索引文件）存储在本地的热数据桶中。这是数据桶的主要副本。

4. 管理器节点为该对等节点提供其复制的数据的目标对等节点的列表。例如，如果复制因子为 3，则管理器节点为该对等节点提供两个目标对等节点的列表。

5. 如果搜索因子大于 1，管理器节点还会告知该对等节点它的哪个目标对等节点应将其数据副本设为**可搜索**副本。例如，如果搜索因子为 2，则管理器节点将指定一个应将其副本设为可搜索副本的特定目标对等节点，并将此信息传达给数据来源对等节点。

6. 该对等节点开始以流化方式将已处理的原始数据传送到管理器节点所指定的目标对等节点。它不会一直等到原始数据文件完成后才开始传送其内容，而是会在处理传入数据的同时以块的形式流送原始数据。如果任何目标对等节点需要将其副本设为可搜索副本（如同步骤 5 中管理器节点与其沟通的情况），该对等节点还会向其发送通知。

7. 目标对等节点从数据来源对等节点接收原始数据并将其存储在本地数据桶副本中。

8. 含有指定可搜索副本的任何目标开始创建必要的索引文件。

9. 该对等节点继续以流化方式将数据传送到目标，直到它滚动其热数据桶。

注意：数据来源对等节点和目标对等节点很少会通过各自的管理端口相互通信。通常，它们只通过复制端口相互间收发数据。管理器节点会管理整个过程。

这只是从单个对等节点流出的明细数据。在群集中，多个对等节点将随时发出和接收数据。

当对等节点滚动热数据桶时

当数据来源对等节点将热数据桶滚动到温数据桶（例如，因为数据桶已经达到其最大大小），会发生以下一系列事件：

1. 数据来源对等节点将其已经滚动数据桶的情况通知给管理器节点及其目标对等节点。

2. 目标对等节点滚动各自的数据桶副本。

3. 在此过程发生期间，数据来源对等节点继续获取外部数据。它会在新的热数据桶中建立数据的本地索引，并以流化方式将其从管理器节点获取的原始数据传送到一组新的目标对等节点。

4. 这组新的目标对等节点从数据来源对等节点接收新热数据桶的原始数据，并将其存储在本地数据桶副本中。含有指定可搜索副本的目标也开始创建必要的索引文件。

5. 数据来源对等节点继续以流化方式将数据传送到目标，直到它滚动其下一个热数据桶。以此类推。

对等节点如何与转发器交互

对等节点从转发器获得其数据时，它会依照任何索引器从转发器获取数据时的方式来处理数据。但是，在群集化环境中，通常应该在向对等节点发送数据的每个转发器上启用**索引器确认**。这可以防止转发器与对等节点之间丢失数据，也是确保端到端数据保真度的唯一方法。如果转发器未获得它已向对等节点发送的数据块的确认，它会重新发送数据块。

有关如何将转发器设置为向对等节点发送数据的详细信息，请阅读“使用转发器将数据导入索引器群集”。要了解对等节点和转发器如何处理索引器确认，请阅读该主题中的“索引器确认如何工作”部分。

在索引器群集中搜索如何工作

在单站点索引器群集中，搜索头在整个对等节点集上执行搜索。

有了多站点索引器群集，您可以实现**搜索相关性**。有了搜索相关性，搜索会在搜索头所在站点的节点之间进行。这样，就提高了网络效率，而没有降低对完整的群集数据集合的访问。

在极少数情况下（参阅下文），您可能想要对单个对等节点执行搜索。

在单个站点群集之间搜索

在索引器群集间搜索的工作方式类似于**分布式搜索**在非群集索引器上的工作方式。主要区别在于，**搜索头**将从管理器节点获取**搜索对等节点**的列表。此外，还会从管理器节点获取生成期间 ID。之后，它将直接与对等节点进行通信。

注意：在索引器群集搜索中，搜索对等节点是当前已注册到管理器节点（换言之，已启动运行且参与到群集的对等节点）的一组群集对等节点。

搜索头启动搜索时：

1. 该搜索头将与管理器节点接触。

2. 管理器节点为搜索头提供当前生成期间 ID 以及该生成期间中的对等节点（即当前已注册到管理器节点的对等节点）的列表。

3. 搜索头与搜索对等节点的通信方式与其在不包含索引器群集的分布式搜索中的方式是一样的。除了为搜索对等节点提供生成期间 ID 之外，还提供完全一样的信息给这些对等节点（搜索请求和知识软件包）。

4. 搜索对等节点使用生成期间 ID 来识别哪个数据桶副本（如果有）是该生成期间的**主要**副本，因此需要参与到搜索中。与在任何其他搜索中相同，对等节点还将使用搜索的时间范围来确定是否搜索特定数据桶。

5. 搜索对等节点对其主要数据桶副本进行搜索并将结果发送回搜索头，之后搜索头负责对结果进行合并。

为了搜索头的可扩展性和高可用性，您可以集成索引器群集和搜索头群集。请参阅《*分布式搜索*》手册中的“集成搜索头群集和索引器群集”。

有关分布式搜索的这些功能和其他可用功能的详细信息，请阅读*《分布式搜索》*手册，从“关于分布式搜索”开始阅读。另请阅读本手册中的“配置搜索头”，了解处理索引器群集中搜索头时的一些配置差异。

搜索和多站点群集

在多站点群集中搜索功能的方式取决于是否针对搜索相似性配置了搜索头。

在多站点群集中进行本地搜索

在多站点群集中，通常会在每个站点放搜索头。这样就可以利用搜索相关性。在搜索相关性中，通常，搜索只会返回来自搜索头所在站点上的对等节点的结果。

默认情况下，多站点群集上的搜索头已启用搜索相似性。然而，您必须执行一些步骤来充分利用。具体地说，您必须确保可搜索数据和搜索头都能在本地获得。关于如何设置搜索相关性的信息，请参阅“在多站点索引器群集中实现搜索相关性”。

一旦站点配置了搜索相关性，则实际搜索过程和单个站点群集一样。搜索头分发当前的生成期间 ID 以及搜索和知识软件包给整个群集中的所有节点。然而，如果群集处于**有效**状态，仅有本地对等节点会响应。本地节点搜索它们的主要数据桶并返回结果到搜索头，使用生成期间 ID 确定哪些数据桶副本是主要副本。

如果群集不处于有效状态并且本地站点没有完全补足主要副本（通常是由于站点上的一些对等节点有故障），则远程节点也会参与搜索，提供本地对等节点缺失的所有主要副本的结果给站点。在这种情况下，搜索不遵守搜索相关性，以保持对整个数据组的访问。一旦站点回到有效状态，接下来的搜索会再次遵守搜索相关性。

注意：热数据桶数据在数据桶中进行复制，如“群集式索引如何工作”中所述。如果本地搜索包括一份复制的热数据桶副本（原始副本在另一个不同的站点上），则在本地节点等待从原始节点获取最新热数据块时，可能会有时间延迟。在此期间，搜索不会返回最新数据。

在多站点群集中进行全局搜索

如果禁用了搜索头的搜索相似性（将其站点设置为 "site0"），则搜索头会使用 site0 主要副本集，通常包含群集内所有站点的主要副本。site0 主要副本集是从每个站点上的可搜索副本中随机选择的，因此 bucketA 的 site0 主要副本可能在 site1 上，bucketB 的 site0 主要副本可能在 site2 上，依此类推。

尽管 site0 主要副本的选择结果在功能上是随机的，但任何数据桶的 site0 主要副本一开始都是作为数据桶在其源站点上的主要副本。随着时间的推移，site0 主要副本可能会因主要副本重新平衡和群集重启之类的操作而变为其他（目标）站点的主要副本。

有关禁用搜索相似性的信息，请参阅“禁用搜索相似性”。

搜索单个对等节点

在进行调试时，您可能偶尔需要搜索单个对等节点。为此，您可以按正常方式在对等节点上直接启动搜索。搜索将访问该对等节点上的所有**可搜索**数据，而不会访问该对等节点上不可搜索的数据副本或其他对等节点上的可搜索数据副本。

注意：请牢记，无法专门将个别对等节点上的某一部分数据配置为可搜索数据。但是，至少通过对等节点进入群集的所有数据都应在该对等节点上可搜索。

索引器群集如何处理报表和数据模型加速摘要

默认情况下，索引器群集不会复制**报表加速**和**数据模型加速**摘要。这意味着只有主要数据桶副本具有关联摘要。

您可以配置管理器节点，使群集复制摘要。之后，所有可搜索数据桶副本即会具有关联摘要。**建议采用这种行为。**

注意：版本为 6.3 或以下的对等节点没有复制摘要功能。

有关报表加速和数据模型加速的详细信息，请参阅*《知识管理器手册》*中的“使用数据摘要加速搜索”一章。

关于摘要和 SmartStore 的信息，请参阅“SmartStore 如何处理报表和数据模型加速摘要”。

摘要的驻留位置

摘要驻留在各自目录相应的对等节点上。在 indexes.conf 中指定目录位置，其中 summaryHomePath 和 tstatsHomePath 属性分别用于报表加速和数据模型加速摘要。详细信息请参阅 indexes.conf 规范文件。

摘要会与一个或多个数据桶相关，这取决于摘要的时间跨度。

复制的摘要

如果要使群集复制摘要，则必须在管理器节点的 server.conf 文件中设置此属性：

```
[clustering]
summary_replication = true
```

必须重新启动管理器节点。

也可以在管理器节点上使用 CLI 命令来设置此属性：

```
splunk edit cluster-config -summary_replication true
```

此命令不需要重新启动。

当群集配置为复制摘要时，该群集会执行一些步骤，确保每个可搜索数据桶副本均包含一个关联摘要副本：

- **对于热数据桶。**群集为热数据桶的每个可搜索副本创建一个摘要。
- **对于温/冷数据桶。**群集在需要时为温或冷数据桶的可搜索副本复制摘要。该群集会通过复制来为温或冷数据桶的可搜索副本补充任何丢失的摘要。

第一次启用摘要复制时，群集可能需要复制大量的摘要。这可能会对网络带宽造成影响。为限制并发摘要复制的数量，可以修改 `max_peer_sum_rep_load` 属性（位于管理器节点的 `server.conf` 文件中）的值。其默认值为 5。

非复制的摘要

如果采用默认行为，群集则不会复制摘要。本节描述群集如何处理非复制的摘要。

摘要会与一个或多个数据桶相关，这取决于摘要的时间跨度。当摘要生成时，它驻留在该时间跨度内拥有数据桶主要副本的对等节点上。如果摘要跨越多个数据桶，并且这些数据桶的主要副本驻留在多个对等节点上，那么这些对等节点中的每一个都将拥有摘要的相应部分。

如果**主要性**从数据桶的一个副本重新分配到另一个副本（例如，由于拥有主要副本的对等节点发生故障），则摘要不会移动到拥有新主要副本的对等节点。因此，它会变得不可用。直到下一次 Splunk Enterprise 尝试更新摘要，发现它已丢失，然后重新生成它，摘要才再次可用。

在多站点群集中，像单个站点群集一样，摘要与主要的数据桶副本一起驻留。因为多站点群集中有多个主要副本，每个支持搜索相关性的站点都有一个，摘要与特定的主要副本（运行搜索时生成搜索头会访问该主要副本）一起驻留。由于站点具有相关性，这通常意味着摘要驻留在与生成搜索头同一站点的主要副本上。

摘要复制和资源争用

启用了加速的搜索头会在对等节点上运行特殊搜索。这些搜索会构建摘要。例如，可参阅*《知识管理器手册》*的“管理报表加速”中有关构建报表加速摘要的描述。

如果是索引器群集上的复制的摘要，热数据桶的每个可搜索副本都会构建摘要。一个对等节点可以同时为源自该对等节点的数据桶副本以及源自其他对等节点的数据桶副本构建摘要。这意味着，在启用摘要复制后，生成摘要的搜索会占用群集内的更多资源，并需要更长时间才能完成。

索引器群集节点如何启动

本主题介绍执行以下操作时发生什么情况：

- 管理器节点启动时
- 对等节点加入新群集时
- 对等节点加入现有群集时

管理器节点启动时

当管理器节点联机（首次联机或再次联机）时，它开始侦听群集对等节点。每个联机对等节点向管理器节点注册，管理器节点会将其添加到群集中。管理器节点会等到复制因子数量的对等节点注册完毕，然后才开始执行其功能。

当您首次部署群集时，您必须首先启用管理器节点，然后启用对等节点，如“索引器群集部署概述”所述。管理器节点将阻止在对等节点上建立索引，直到您启用并重新启动了所有复制因子数量的对等节点为止。

后续重新启动管理器节点时，它会安静地等待一段时间，以便所有对等节点有时间向其注册。安静等待时间结束后，向其注册的对等节点数已达到复制因子指定的数量，管理器节点即可开始执行协调功能，例如，重新平衡**主要**数据桶副本，以及告知对等节点将传入数据的副本流送到何处。因此，在您重新启动管理器节点时，您必须确保至少有复制因子指定数量的对等节点正在运行。

一段时间结束后，您可以查看管理器节点仪表板了解群集状态的信息。

有关管理器节点发生故障并重新启动时发生事件的更多信息，请参阅“管理器节点故障时的情况”。

对等节点加入新群集时

当您首次部署群集时，必须首先启用管理器节点，然后启用对等节点，如“索引器群集部署概述”中所述。管理器节点将阻止在对等节点上建立索引，直到您启用并重新启动了所有复制因子数量的对等节点为止。

每个对等节点在联机时向管理器节点注册，管理器节点会将最新的**配置软件包**自动分发给它。该对等节点会在本地对配置软件包进行验证。只有软件包验证成功时，对等节点才会加入群集。

在加入群集的对等节点数达到复制因子指定数量后，对等节点就会开始为数据建立索引。

对等节点加入现有群集时

对等节点也可以在后来的某个时间进行联机，也就是在群集已经运行并具有一个管理器节点和复制因子数目的对等节点的情况下。对等节点联机时将向管理器节点进行注册，管理器节点会将最新的配置软件包分布到该对等节点。该对等节点会在本地对配置软件包进行验证。只有软件包验证成功时，对等节点才会加入群集。

注意：添加新对等节点到现有群集将导致跨一组现有对等节点重新平衡主要数据桶副本，如“重新平衡索引器群集的主要数据桶”所述。然而，新对等节点不会获得任何主要副本，因为管理器节点仅在一组可搜索副本上执行重新平衡，同时新对等节点在启动时没有任何可搜索副本。该对等节点可以参与将来的数据桶复制，但管理器节点不会自动将现有对等节点的数据桶副本转移到新对等节点上，也不会将现有对等节点的主要副本重新分配到新对等节点。

对等节点故障时的情况

对等节点故障可能是有意（通过调用 CLI 命令 `offline`，如“使对等节点脱机”中所述）行为，也可能是无意（例如，由于服务器崩溃）行为。

无论对等节点的故障原因为何，管理器节点都将协调补救活动，以重新创建一组完整的补充数据桶副本。此过程称为**数据桶修复**。管理器节点将跟踪每个节点上包含哪些数据桶副本以及这些副本的状态（**主要性**、**可搜索性**）。因此，当某一对等节点故障时，管理器节点可以指示剩余对等节点修复群集的数据桶集合，以便恢复群集的状态：

- 每个数据桶正好一个**主要**副本（**有效**状态）。在多站点群集中，有效状态意味着：基于 `site_search_factor`，在每个支持搜索相关性的站点上有一个主要副本。
- 每个数据桶的完整**可搜索**副本集（与搜索因子匹配）。对于多站点群集的情况，可搜索数据桶副本的数量也必须满足站点特定的搜索因子的需要。
- 每个数据桶的包含可搜索和不可搜索副本在内的完整副本集，与复制因子（**完整**状态）匹配。对于多站点群集的情况，数据桶副本的数量也必须满足站点特定的复制因子的需要。

除了多个节点的灾难性故障以外，管理器节点通常可以重新创建有效群集。如果群集（或多站点群集中的站点）的搜索因子至少为 2，则它几乎可以立即执行此操作。管理器节点是否还能够重新创建完整群集，这取决于与复制因子相比，仍然在运行的对等节点的数量。仍在运行的节点数必须至少与复制因子相等，群集才能达到完整状态。在多站点群集的情况下，每个站点必须有的可用节点数量至少为站点复制因子所指定的值。

群集可能需要很长的时间才能恢复到完整状态，因为它必须先将数据桶从一个对等节点流送到另一个，并将不可搜索的数据桶副本设为可搜索副本。有关更多信息，请参阅“评估对等节点取消配置时群集的恢复时间”。

除了修复数据桶所执行的补救步骤之外，节点故障时还会发生一些其他重要事件：

- 管理器节点滚动**生成期间**并创建一个新的**生成期间 ID**，在需要时，它会将此 ID 传送给对等节点和搜索头。
- 所有包含故障节点的热数据桶副本的对等节点都会将滚动到温数据桶。

当有意使对等节点脱机时

`splunk offline` 命令将节点从群集删除，然后停止该节点。该命令使对等节点正常关闭，允许所有正在进行的搜索完成，同时也使群集快速恢复到完全可搜索状态。

`splunk offline` 命令有两个版本：

- `splunk offline`：这是 `splunk offline` 命令的快速版。对等节点会很快关闭，所需时间最多不超过五分钟，即使搜索或补救活动仍处于运行状态。
- `splunk offline --enforce-counts`：这是该命令的 `enforce-counts` 版，用于验证群集已返回到完整状态。若调用了 `enforce-counts` 标记，则对等节点在所有补救活动均已完成后才会关闭。

有关运行 `splunk offline` 命令的详细信息，请参阅“使对等节点脱机”。

快速版的脱机命令

快速版 `splunk offline` 命令的语法如下：

```
splunk offline
```

此版本的命令将执行下面的一系列操作：

1. **部分关闭。**对等节点立即执行部分关闭。对等节点停止接受外部输入和复制的数据。它暂时会继续参与搜索。

2. **主要性重新分配。**管理器节点将对等节点上的任何主要数据桶副本的主要性重新分配给其他对等节点上的这些数据桶可用的可搜索副本（如果是多站点群集，则在同一站点上）。在此步骤结束时（如果群集或群集站点的搜索因子至少为 2，则此过程会花费一些的时间），群集将恢复到有效状态。

此步骤执行过程中，对等节点的状态为 `ReassigningPrimaries`。

3. **生成期间 ID 滚动。**管理器节点滚动群集的生成期间 ID。在此步骤结束时，对等节点不再参与新搜索，但会继续参与所有正在执行的搜索。

4. **完全关闭。**在最多不超过五分钟后或正在执行的搜索和主要性重新分配活动完成时（以先到者为准），对等节点将完全关闭。它不再向管理器节点发送检测信号。有关完全关闭之前需具备条件的详细信息，请参阅“快速脱机过程”。

5. **重新启动计数。**对等节点关闭后，管理器节点将按照 `restart_timeout` 属性所定义的时间长度（默认为 60 秒，在 `server.conf` 中设置）进行等待。如果对等节点在此时间段内重新联机，则管理器节点将重新平衡主要数据桶副本的群集集合，但不执行进一步的补救活动。

此步骤执行过程中，对等节点的状态为 `Restarting`。如果对等节点在此超时期间内未重新联机，其状态会变为 `Down`。

6. **补救活动。**如果对等节点在 `restart_timeout` 期间内未重新启动，管理器节点将执行补救操作来修复群集数据桶并将群集返回至完整状态。它将告知剩余对等节点将脱机对等节点上的数据桶副本复制到其他对等节点上。此外，还会通过指示其他对等节点将这些数据桶的不可搜索副本设为可搜索副本，来补偿脱机对等节点上的所有可搜索数据桶副本。在此步骤结束时，群集将恢复到完整状态。

如果对等节点脱机后导致剩余节点数小于复制因子，则群集将无法完成此步骤，也无法恢复到完整状态。有关数据桶修复如何运行的详细信息，请参阅“数据桶修复方案”。

在多站点群集的情况下，补救活动尽可能在脱机节点所在的站点中进行。详细信息请参阅“在多站点群集中修复数据桶”。

enforce-counts 版的脱机命令

`enforce-counts` 版 `splunk offline` 命令的语法如下：

```
splunk offline --enforce-counts
```

此版本的 `splunk offline` 命令只有在满足特定条件时才会运行。特别是，如果对等节点脱机会导致群集中剩余对等节点数小于复制因子，则不会运行该命令。有关运行该命令所需的条件集，请参阅 `enforce-counts` 版的脱机过程。

此命令版本启动一个名为**取消配置**的过程，此过程中将执行下面的一系列操作：

1. **部分关闭。**对等节点立即执行部分关闭。对等节点停止接受外部输入和复制的数据。它暂时会继续参与搜索。

2. **主要性重新分配。**管理器节点将对等节点上的任何主要数据桶副本的主要性重新分配给其他对等节点上的这些数据桶可用的可搜索副本（如果是多站点群集，则在同一站点上）。在此步骤结束时（如果群集或群集站点的搜索因子至少为 2，则此过程会花费一些的时间），群集将恢复到有效状态。

此步骤执行过程中，对等节点的状态为 `ReassigningPrimaries`。

3. **生成期间 ID 滚动。**管理器节点滚动群集的生成期间 ID。在此步骤结束时，对等节点不再参与新搜索，但会继续参与所有正在执行的搜索。

4. **补救活动。**管理器节点执行补救操作来修复群集数据桶以便群集恢复到完整状态。它将告知剩余对等节点将脱机对等节点上的数据桶副本复制到其他对等节点上。此外，还会通过指示其他对等节点将这些数据桶的不可搜索副本设为可搜索副本，来补偿脱机对等节点上的所有可搜索数据桶副本。在此步骤结束时，群集将恢复到完整状态。

此步骤执行过程中，对等节点的状态为 `Decommissioning`。有关数据桶修复如何运行的详细信息，请参阅“数据桶修复方案”。

5. **完全关闭。**当群集恢复到完整状态时对等节点关闭。一旦关闭，此对等节点无法再向管理器节点发送检测信号。此时，对等节点的状态变为 `GracefulShutdown`。有关完全关闭之前需具备条件的详细信息，请参阅 `enforce-counts` 版的脱机过程。

当对等节点无意关闭时

当某一对等节点因 offline 命令以外的任何原因关闭时，它会停止向管理器节点发送定期的检测信号。这样，管理器节点便会检测到丢失情况，并启动补救操作。除了以下事件外，管理器节点协调的操作基本上与对等节点有意脱机的情况相同：

- 脱机的对等节点不继续参与正在执行的搜索。
- 在重新分配主要性和启动数据桶修复操作之前，管理器节点只会等待检测信号超时的时间长度（默认为 60 秒）。

在某个节点故障之后，搜索可以继续在群集中执行；但是，搜索将仅提供部分结果，直到群集重新获得其有效状态。

在多站点群集的情况下，当站点上的一个对等节点有故障时，站点会丢失其搜索相关性（如果有的话），直到它重新获得有效状态。在此期间，搜索会通过参与进来的远程节点，继续提供完整的结果。

数据桶修复方案

要更换关闭对等节点上的数据桶副本，管理器节点将协调对等节点之间的数据桶修复活动。除了更换所有数据桶副本之外，群集必须确保它重新获得主要和可搜索副本。

注意：对于具有 **SmartStore** 索引的群集，数据桶修复流程会有些不同。请参阅“索引器群集操作和 SmartStore”。

数据桶修复涉及三种活动：

- 通过分配主要状态给其他对等节点上的这些数据桶的可搜索副本，**补偿关闭对等节点上的任何主要副本**。
- 通过将其他对等节点上的这些数据桶从不可搜索转换为可搜索，**补偿任何可搜索副本**。
- 通过流化每个数据桶的副本到还未拥有该数据桶副本的对等节点，**更换所有数据桶副本**（可搜索和不可搜索）。

例如，假定关闭对等节点具有 10 个数据桶副本，同时其中的五个可搜索，同时两个可搜索副本是主要副本。群集必须：

- 重新分配主要状态给其他对等节点上的两个可搜索副本。
- 将其他对等节点上的五个不可搜索数据桶副本转换为可搜索。
- 从一个活跃对等节点流化 10 个数据桶副本到另外一个。

首次活动 - 将数据桶的可搜索副本从非主要副本转换为主要副本 - 这个速度非常快，因为可搜索数据桶副本已经具有索引文件，因此实际上不会涉及任何处理操作。（这是假定有一个备用的可搜索副本可用，即要求搜索因子至少为 2。如果不是，则群集必须将不可搜索副本变成可搜索副本，然后才能指定它为主要副本。）

第二个活动 - 将数据桶的不可搜索副本转换为可搜索副本需要一些时间，因为对等节点必须从一个可搜索副本复制数据桶的索引文件到另一个对等节点（或者，如果没有该数据桶的任何其他可搜索副本，则对等节点必须从原始数据文件重新构建数据桶的索引文件）。有关评估使不可搜索副本成为可搜索副本所需时间的帮助，请阅读“评估对等节点取消配置时群集的恢复时间”。

第三个活动 - 从一个对等节点流化的版本到另一个需要大量时间（取决于要流化的数据量），如“评估对等节点取消配置时群集的恢复时间”中所述。

下面的两个示例显示了管理器节点 1）如何重新创建有效的完整群集，以及 2）如何在保留的活动节点不足时创建有效非完整群集。无论对等节点是有意还是无意关闭，此过程都是相同的。

请记住以下几点：

- 当每个数据桶有一个主要可搜索副本时，群集为有效。在有效群集中执行的任何搜索都将提供一组完整的搜索结果。
- 当群集中的数据桶副本数等于复制因子且可搜索副本数等于搜索因子时，群集为完整。
- 如果群集包含所有数据桶的可搜索副本，但数据桶副本数量小于复制因子，则群集可以是有效但不完整的群集。因此，如果复制因子为 3 的群集只有刚好三个对等节点且其中一个对等节点故障，则此群集可以成为有效群集，但不能成为完整群集，因为只有两个活动节点时，无法通过保留三组数据桶副本来达到复制因子值。

示例：修复数据桶以创建有效完整群集

假设：

- 对等节点意外关闭（即，未响应 offline 命令）。
- 故障对等节点是具有以下特性的群集的一部分：
  - 5 个对等节点，包括故障的对等节点
  - 复制因子 = 3
  - 搜索因子 = 2
- 故障对等节点包含以下数据桶副本：
  - 数据桶的 3 个主要副本
  - 10 个可搜索副本（包括主要副本）

- 总计 20 个数据桶副本（结合可搜索副本和不可搜索副本）

当对等节点故障时，管理器节点按如下方式将消息发送到各个剩余对等节点：

1. 对于故障节点上的三个主要数据桶副本中的每个副本，管理器节点确定包含该数据桶的另一个可搜索副本的节点，并指示该节点将该副本标记为主要副本。

当此步骤完成时，群集会重新获得有效状态，并且任何后续搜索都将提供一组完整的结果。

2. 对于故障节点上的 10 个可搜索数据桶副本中的每个副本，管理器节点会确定 1）含有该数据桶可搜索副本的节点；2）含有同一数据桶不可搜索副本的节点。然后，它会指示含有可搜索副本的节点以流化方式将数据桶的索引文件传送到第二个节点。当索引文件已被复制后，不可搜索副本变成可搜索副本。

3. 对于故障节点上的共计 20 个数据桶副本中的每个副本，管理器节点会确定 1）含有该数据桶副本的节点；2）不含有该数据桶副本的节点。然后，它指示带副本的节点流化数据桶的原始数据到第二个节点，生成该数据桶的全新不可搜索副本。

当这最后一步完成时，群集将重新获得完整状态。

示例：修复数据桶以创建有效不完整群集

假设：

- 对等节点意外关闭（即，*未*响应 `offline` 命令）。
- 故障对等节点是具有以下特性的群集的一部分：
  - 3 个对等节点，包括故障的对等节点
  - 复制因子 = 3
  - 搜索因子 = 1
- 故障对等节点包含以下数据桶副本：
  - 数据桶的 5 个主要副本
  - 5 个可搜索副本（与主要副本的数量相同；因为搜索因子 = 1，所有可搜索副本必须都是主要副本。）
  - 总计 20 个数据桶副本（结合可搜索副本和不可搜索副本）

由于群集只有三个节点且复制因子为 3，如果出现故障节点，则意味着群集无法再达到复制因子值，因此无法成为完整群集。

当对等节点故障时，管理器节点按如下方式将消息发送到各个剩余对等节点：

1. 对于故障节点上的五个可搜索主要数据桶副本中的每个副本，管理器节点首先确定含有不可搜索副本的节点，并指示该节点使该副本成为可搜索副本。然后，该节点开始为该副本构建索引文件。（由于搜索因子是 1，因此剩余节点上没有这些数据桶的任何其他可搜索副本。因此，无法通过从另一个可搜索副本流化索引文件，让剩余对等节点使得不可搜索数据桶副本变为可搜索。与此相反，它们必须采用更加缓慢的流程，从不可搜索副本的原始数据文件创建索引文件。）

2. 然后，管理器节点指示来自步骤 1 的节点标记 5 个最新的可搜索副本为主要副本。与先前示例不同的是，只有在不可搜索副本被标记为可搜索副本时，才能将其他数据桶副本指定为主要副本。因为群集的搜索因子 = 1，因此没有备用可搜索副本。

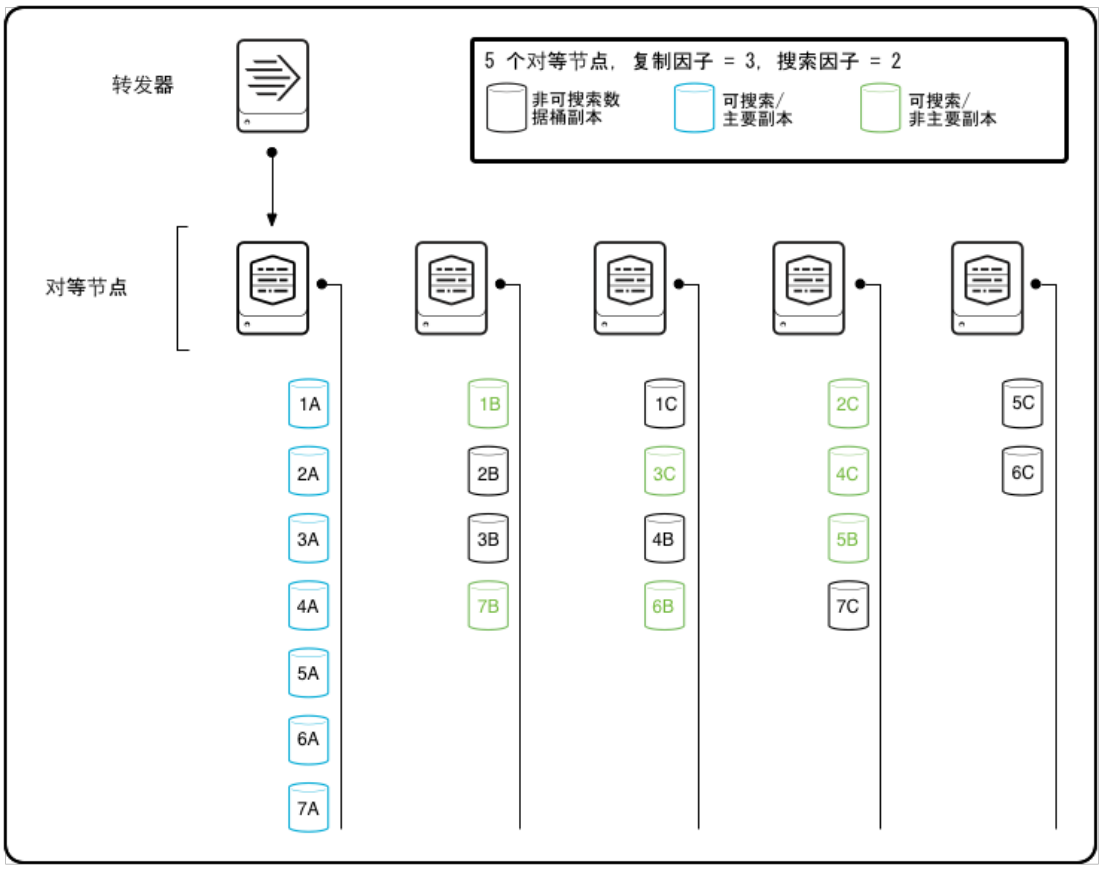
一旦步骤 2 完成，群集重新获得有效状态。任何随后的搜索将提供完整的结果。

3. 对于故障节点上的总计 20 个数据桶副本，管理器节点无法启动任何操作来创建替代副本（以便群集再次拥有每个数据桶的三个副本，如同复制因子指定的那样），因为没有剩下足够多的节点用来保存副本。

由于群集无法重新创建与复制因子相等的一组完整的数据桶副本，因此该群集仍然处于不完整状态。

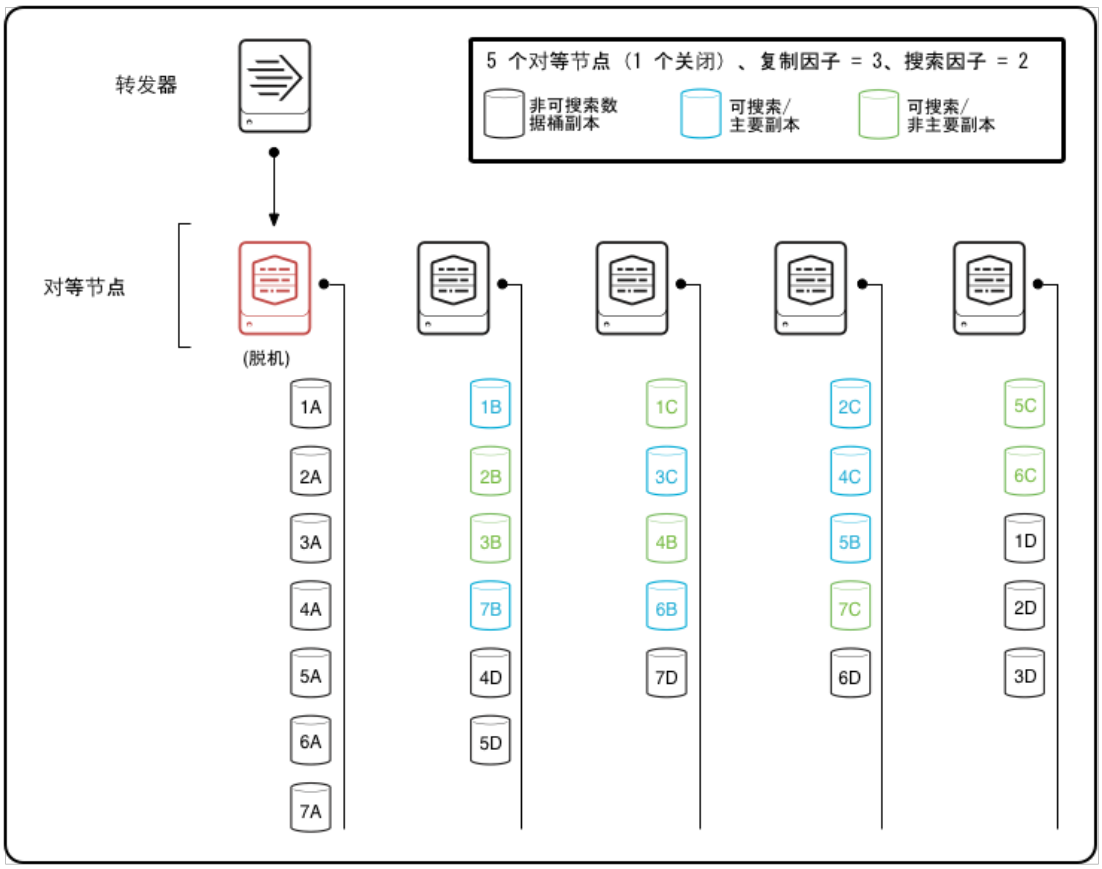
图片

下图显示一个有五个对等节点的群集，其复制因子为 3，搜索因子为 2。主要数据桶副本驻留在从转发器接收数据的数据来源对等节点上，该数据的可搜索副本和不可搜索副本分散在其他对等节点上。



注意：此图是高度简化的图表。为减少复杂性，只显示了来自一个对等节点的数据所对应的数据桶。在现实方案中，其他大多数的对等节点（如果不是所有）也会成为原始数据并复制到群集上的其他对等节点。

下图显示了同样的该群集的简化版本，在保留所有主要副本的数据来源节点发生故障后，管理器节点已经发出了让剩余对等节点修复数据桶的指令：



管理器节点通过执行一系列活动引导群集从故障对等节点的情景中恢复回来：

1. 管理器节点将故障对等节点上的数据桶副本的主要性重新分配给剩余对等节点上的可搜索副本。完成此步骤后，它滚动生成期间 ID。

2. 主节点指示对等节点将不可搜索副本标记为可搜索副本，以补足缺失的这组可搜索副本。

3. 它指示对一组新的不可搜索副本（1D、2D 等）进行复制，在剩余对等节点之间分散。

即使数据来源节点发生故障，群集拥有的全部数据桶数量仍等于复制因子、可搜索数据桶副本数量等于搜索因子，以及每个数据桶的一个主要副本，仍可以完全恢复其完整和有效状态。此图中所表示的生成期间与上图有所不同，因为主要副本已经移到了不同的对等节点。

在多站点群集中修复数据桶

多站点群集处理节点故障的过程与单个站点群集有一些显著差别。请参阅“多站点群集和节点故障”。

查看数据桶修复状态

您可以在管理器节点仪表板上查看数据桶修复状态。请参阅“查看管理器节点仪表板”。

对等节点重新联机时的情况

对等节点可以有意地（通过 CLI 命令 `offline`）或无意地（例如，由于服务器崩溃）关闭。当对等节点关闭时，群集将执行补救活动，也称为**数据桶修复**，如“对等节点故障时的情况”主题所述。本主题将介绍当对等节点稍后重新返回群集中时会发生什么情况。

对等节点重新回到群集时，它会开始向管理器节点发送检测信号。管理器节点将识别该对等节点并重新将其添加回群集中。如果该对等节点在群集先前活动中的数据桶副本未受影响，则管理器节点会将这些副本添加到它所维护的数据桶计数中。管理器节点还重新平衡群集，这会导致对等节点上的可搜索数据桶副本（如果有）开始分配**主要性**状态。有关重新平衡的信息，请参阅“重新平衡索引器群集的主要数据桶”。

注意：当对等节点与管理器节点连接时，它会检查自己是否已经具有**配置软件包**的当前版本。如果软件包在该对等节点故障后发生了变化，则该对等节点会下载最新的配置软件包，在本地对其进行验证，然后重新启动。只有软件包验证成功时，对等节点才会重新加入群集。

管理器节点如何对数据桶计数

要了解对等节点重新返回群集中会发生什么情况，您必须首先了解管理器节点如何跟踪数据桶副本。

管理器节点会为群集中的每个数据桶维护计数。对于每个数据桶，它会了解：

- 群集中存在数据桶的多少个副本。
- 群集中存在数据桶的多少个**可搜索**副本。

管理器节点还会确保给定数据桶始终拥有正好一个**主要**副本。

在多站点群集中，管理器节点会为每个站点（作为一个整体，也为群集）跟踪副本和可搜索副本。它还确保每个有显式搜索因子的站点有且仅有一份每个数据桶的主要副本。

通过这些计数，管理器节点可以确定群集是否处于**有效**和**完整**状态。对于单个站点群集来说，这意味着群集有：

- 每个数据桶的一个主要副本。
- 每个数据桶的完整可搜索副本集（与搜索因子匹配）。
- 每个数据桶的包含可搜索和不可搜索副本在内的完整副本集（与复制因子匹配）。

对于一个多站点群集来说，一个有效完整群集有：

- 恰好一份（每个带显式搜索因子的站点的每个数据桶的）主要副本。
- 每个数据桶的完整的可搜索副本集合，匹配每个站点（作为一个整体，也匹配群集）的搜索因子。
- 每个数据桶的完整的副本（包括可搜索的和不可搜索的）集合，匹配每个站点（作为一个整体，也匹配群集）的复制因子。

数据桶修复和对等节点上的副本

如果某个对等节点出现故障，管理器节点将指示剩余对等节点参与到数据桶修复活动中。最后，如果数据桶修复成功，群集恢复完整状态。

如果之后对等节点重新返回到群集中，管理器节点会将其数据桶副本数加到计数中（首先假定导致该对等节点关闭的问题并没有损坏这些副本）。所得结果会稍有不同，这取决于对等节点重新返回时数据桶修复活动是否已经完成。

如果数据桶修复已经完成

如果数据桶修复已经完成，群集则处于完整状态，那么返回的对等节点上的副本就是多余的。例如，假定复制因子为 3，并且群集已经修复所有数据桶以使群集中又再次存在每个数据桶的三个副本，包括故障对等节点在关闭之前所保有的那些副本。当故障对等节点之后重新返回并且其副本并未受到影响时，管理器节点只是将这些副本加到计数中，这样，群集中就存在某些数据桶的四个副本，而不是三个。同样，如果返回的对等节点中存在一些可搜索数据桶副本，则就会产生多余的可搜索数据桶副本。当保有这些数据桶中其中一部分副本的另一个对等节点出现故障时，这些多余的副本就会发挥作用。

如果数据桶修复仍在执行中

如果群集仍在替换对等节点故障时所丢失的副本，那么对等节点的重新返回可以取消数据桶修复。管理器节点将恢复联机的对等节点上的副本数加到计数后，就会知道群集完整有效，从而不再指示其他对等节点创建这些数据桶的副本。但是，当前正在执行某些数据桶修复活动（例如，复制数据桶或将副本标记为可搜索副本）的所有对等节点将会完成正在对这些副本执行的操作。由于数据桶修复相当耗时，因此最好是尽快让故障的对等节点恢复联机，尤其是在对等节点上保有大量数据桶副本的情况下。

删除多余的数据桶副本

如果返回的对等节点导致一些数据桶的额外副本，您可以删除额外副本以节省磁盘空间。请参阅“从索引器群集中删除多余的数据桶副本”。

管理器节点故障时的情况

管理器节点对于正常运行索引器群集是非常重要的，它将作为大部分群集活动的协调者。但是，如果管理器节点发生故障，对等节点和搜索头具有默认的行为，这使它们至少在一段时间内可以相当正常的运行。尽管如此，您仍然应将管理器节点故障视为严重故障。

为了应付可能出现的管理器节点故障，可以配置备用管理器节点，如果需要可以接管。有关详细信息，请参阅“在索引器群集中替换管理器节点”。

当管理器节点发生故障时

如果管理器节点发生故障，群集可以继续正常运行，只要没有发生其他故障。对等节点可以继续插入数据，将副本流送到其他对等节点，复制数据桶并对来自搜索头的搜索请求做出响应。

当对等节点滚动热数据桶时，它通常会联系管理器节点以获得要将其下一个热数据桶流送到的目标对等节点的列表。但是，如果对等节点在管理器节点故障时滚动热数据桶，它就会将其下一个热数据桶开始流送到用作其上一个热数据桶的目标的同一组对等节点。

最终，将开始出现问题。例如，如果对等节点发生故障，而管理器节点仍旧故障，则无法协调必要的补救**数据桶修复**活动。或者，如果由于某些原因，对等节点无法与其一个目标对等节点建立连接，那么它将无法获得另一个目标。

搜索头也可继续执行而没有管理器节点，尽管最终这些搜索很可能访问的是不完整的一组数据。（例如，如果带主要数据桶副本的对等节点出现故障，将没有办法将主要性转移到其他对等节点上的副本，因此这些数据桶将不再可用访问。）搜索头将使用它在管理器节点故障之前所获得的最后生成期间 ID。如果最后生成期间中的一个或多个对等节点发生故障，它会显示一条警告。

管理器节点重新启动时

对等节点继续无限期地发送检测信号，以便管理器节点重新启动时，它们可以检测到管理器节点并重新连接。

当管理器节点重新启动时，它会安静地等待一段时间，以便所有对等节点有时间重新向其注册。在这段时间结束后，管理器节点将获得群集状态的完整视图，包括对等节点以及数据桶的状态。假设至少复制因子数目的对等节点已向管理器节点注册，管理器节点会发起任何需要的数据桶修复活动，以确保群集**有效**且**完整**。此外，它还将根据需要重新平衡群集和更新生成期间 ID。

数据桶修复需要一些时间完成，因为它涉及复制数据桶并让不可搜索副本变为可搜索。有关完成数据桶修复活动所需时间的帮助，请查看此处。

一段时间结束后，可以查看管理器节点仪表板了解群集状态的准确信息。

注意：在您重新启动管理器节点时，您必须确保至少有复制因子指定数量的对等节点正在运行。

实现 SmartStore 以减少本地存储要求

关于 SmartStore

SmartStore 是一个索引器功能，使用远程对象存储（例如 Amazon S3 或 Google GCS）存储索引数据。

随着部署数据量的增加，通常存储需求会高于对计算资源的需求。SmartStore 允许您通过单独扩大资源范围，以一种经济有效的方式管理索引器存储和计算资源。

SmartStore 引入了远程存储层和**缓存管理器**。这些功能允许数据驻留在本地索引器或远程存储层中。索引器和远程存储层之间的数据移动由驻留在索引器上的缓存管理器管理。

借助 SmartStore，您可以最大程度地降低索引器存储大小并选择 I/O 优化计算资源。大多数数据驻留在远程存储上，而索引器保留包含最少数据量的本地缓存：热数据桶、参与活动或最近搜索的温数据桶副本以及数据桶元数据。

您可为所有索引或索引子集启用 SmartStore。

SmartStore 优势

SmartStore 在部署的索引层方面有几个优势：

- 降低存储成本。部署可利用远程对象存储的经济性，而不依赖昂贵的本地存储。
- 通过远程对象存储访问可用的高可用性和数据复原功能。
- 此功能能够分别扩展计算和存储资源，从而确保您有效地使用资源。
- 使用各索引设置进行简单灵活的配置。
- 允许新群集或独立索引器继承旧群集或独立索引器数据的启动功能。

SmartStore 在索引器群集部署方面也有几项优势：

- 快速从对等节点故障中恢复，快速重新平衡数据、仅需要修复温数据桶的元数据。
- 总体的存储要求更低，因为系统只保留每个温数据桶的单个永久副本。
- 即使故障的对等节点数大于或等于复制因子，也可以完全恢复温数据桶。
- 全局基于大小的数据保留。
- 简化升级。

对于大部分搜索使用案例来说，智能缓存管理器可确保 SmartStore 提供和本地存储配置类似的性能。

选择 SmartStore

虽然启用 SmartStore 的索引可在适当的情况下明显降低存储和管理成本，您有时也可能会发现继续依赖本地存储更为可取。

考虑移动到 SmartStore 时

SmartStore 可帮助您为中型到大型部署节约大量成本。尤其是，在以下情况下考虑启用 SmartStore：

- 因为本地存储中的数据量会继续增加。虽然本地存储成本可能不是小型部署的重要问题，但是由于部署会随着时间的推移而扩大，您应该重新考虑使用本地存储。
- 如果您正在使用索引器群集以利用数据恢复和灾难恢复等功能。通过 SmartStore，您可通过远程存储的本地功能实现这些目标，不需要在本地存储上存储大量的冗余数据。
- 如果您正在使用索引器群集，并且发现大量的时间和计算资源用于管理群集。通过 SmartStore，您可消除大部分群集管理开销。特别是，您可极大减少耗时活动的规模，如脱机对等节点、数据重新平衡和数据桶修复，因为大部分数据不再驻留在对等节点上。
- 当大部分搜索涉及最近数据时。

不移动到 SmartStore 时

几种情况下，选择本地存储可能会更好：

- 如果您的部署较小，存储的数据量有限，SmartStore 的优势可能无法弥补设置和维护远程存储的成本。
- 如果您经常需要运行少量的搜索，SmartStore 可能不适合您，因为少量的搜索可能要求索引器将大量数据从远程存储复制到本地存储，从而影响性能。对于时间跨度较长的搜索尤为如此。但是，如果搜索是跨最近数据且必要的数据桶已经驻留在缓存中，那么就不会影响性能。
- 如果您经常运行长时间回顾搜索，您可能需要增加缓存大小或继续依赖本地存储。

SmartStore 不支持的功能

以下功能不适用于启用 SmartStore 的索引。相应的设置必须使用默认值。

- Tsidx 减少。请勿将 enableTsidxReduction 设为 "true"。Tsidx 减少会修改数据桶内容，且 SmartStore 不支持。**注意：**在迁移到 SmartStore 之前，您仍然可以搜索任何已减少 tsidx 的现有存储桶。和非 SmartStore 部署一样，这类搜索可能运行缓慢。请参阅“减少 tsidx 磁盘使用”。
- 数据完整性控制功能。启用 SmartStore 的索引和数据完整性控制功能不兼容，详见*确保 Splunk Enterprise 安全*手册中“管理数据完整性”介绍。
- 禁用布隆过滤器。请勿将 createBloomfilter 设为 "false"。布隆过滤器通过减少从远程存储中下载 tsidx 文件在 SmartStore 中起到重要作用。
- 更改布隆过滤器位置。请不要更改 bloomHomePath。布隆过滤器必须保留在数据桶目录中的默认位置。
- 摘要复制。SmartStore 不需要摘要复制，因为创建摘要后会上载至远程存储，且群集中的所有对等节点均可访问。
- Hadoop Data Roll。
- 某些其他 indexes.conf 设置和 SmartStore 不兼容。请参阅“indexes.conf 中和 SmartStore 不兼容或有其他方面限制的设置”

当前 SmartStore 使用限制

此时，SmartStore 支持需要您的索引层符合特定限制条件：

- 对于索引器群集，复制因子和搜索因子必须相等（例如 3/3 或 2/2）。
- 每个索引的主路径和冷路径必须指向相同分区。
- 某些其他 indexes.conf 设置受 SmartStore 限制。请参阅“indexes.conf 中和 SmartStore 不兼容或有其他方面限制的设置”
- 启用 SmartStore 的索引无法转换为非 SmartStore。
- 对于多站点群集，如果任何 SmartStore 索引使用报表加速或数据模型加速，您必须通过将所有搜索头设为 site0 禁用搜索相关性。请参阅“在多站点索引器群集中执行搜索相关性”。

SmartStore 和 Splunk Enterprise Security

SmartStore 可与 Splunk Enterprise Security 5.3.0 及更高版本兼容。

对于 SmartStore 和 Splunk Enterprise Security 一起使用，确认您有足够的本地存储可用于容纳 90 天的索引数据，而不是建议的 30 天。请参阅“本地存储要求”。

SmartStore 架构概述

SmartStore 功能的架构目标在于尽量减少本地存储中的数据量，同时保持 Splunk Enterprise 部署快速索引和搜索功能的特性。除少数不常见的情况，索引器会返回启用 SmartStore 索引的搜索结果，其速度与非 SmartStore 索引的速度类似。

Nutshell 中的 SmartStore

启用 SmartStore 索引最大限度地减少了对本地存储的使用，其大部分数据驻留在远程对象存储（如 S3 或 GCS）上。

在索引器上索引和搜索数据，就像在本地存储所有数据的传统部署一样。

与 SmartStore 的主要区别在于远程对象存储成为温数据桶主要副本的位置，而索引器的本地存储用于缓存当前参与搜索或者很有可能参与未来搜索的温数据桶副本。

索引器上的缓存管理器可从远程存储中提取温数据桶副本，然后在搜索需要数据桶时将数据桶放入索引器本地缓存中。一旦再次搜索的可能性减少，缓存管理器也会从缓存中逐出温数据桶副本。

数据桶和 SmartStore

对于 SmartStore 索引，和非 SmartStore 索引一样，在索引器的本地存储缓存中构建热数据桶。但是对于 SmartStore 索引，当数据桶从热状态滚动到温状态时，会将数据桶副本上载至远程存储，然后远程副本会变成数据桶的主副本。

最后，缓存管理器会逐出缓存中的本地数据桶副本。当索引器需要搜索没有本地副本的温数据桶时，缓存管理器会从远程存储器中提取副本并将其放在本地缓存中。

远程存储有每个温数据桶的副本。

各索引器的本地缓存包含各种类型的数据：

- 热数据桶。在本地存储中创建热数据桶。这些数据桶会继续只驻留在索引器中，直到滚动为温数据桶。
- 目前参与搜索的温数据桶的副本。
- 最近创建或搜索的温数据桶的副本。索引器包含温数据桶缓存，以尽量减少重复从远程存储中提取相同数据桶的需要。
- 远程数据桶的元数据。索引器保留有关远程存储中每个数据桶的少量信息。

SmartStore 索引的数据桶最初只有两种活动状态：热数据桶和温数据桶。冷状态是和非 SmartStore 索引结合使用以区分符合移除廉价存储资格的旧数据，没必要和 SmartStore 结合使用，因为温数据桶已经驻留在廉价远程存储上。数据桶直接从温状态滚动到冻结状态。

事实上，冷数据桶可存在于某种启用 SmartStore 的索引中，但是只在限定情形下。尤其是，如果您将某个索引从非 SmartStore 迁移到 SmartStore，迁移后任何已迁移的冷数据桶使用现有的冷路径作为其缓存位置。

冷数据桶各方面功能和温数据桶相同。缓存管理器可用和管理温数据桶相同的方式管理已迁移的冷数据桶。惟一的区别在于冷数据桶将提取到冷路径位置而不是主路径位置。

缓存管理器

索引器的缓存管理器管理本地缓存。当搜索需要数据桶时会从远程存储中提取温数据桶的副本。还会根据各种因素，如数据桶的搜索频率、数据新近程度和各种其他的可配置条件，将数据桶从缓存中逐出。

绝大多数 Splunk 平台搜索所共有的某些特性会驱动缓存管理器优化数据桶位置的策略。尤其是大部分搜索具有以下特征：

- 针对近期数据进行搜索。97% 的搜索都会回溯到过去 24 小时或更短时间。
- 具有空间和时间局限性。如果搜索在特定时间或特定日志中发现事件，那么其他搜索可能会在相似的时间范围内或该日志中查找事件。

因此，缓存管理器支持最近创建的存储桶和访问的存储桶，确保您可能会搜索的大多数数据在本地缓存中可用。那些很可能偶尔参与搜索的存储桶仍然存在于远程存储中，并且可根据需要提取到本地存储器。

索引器群集

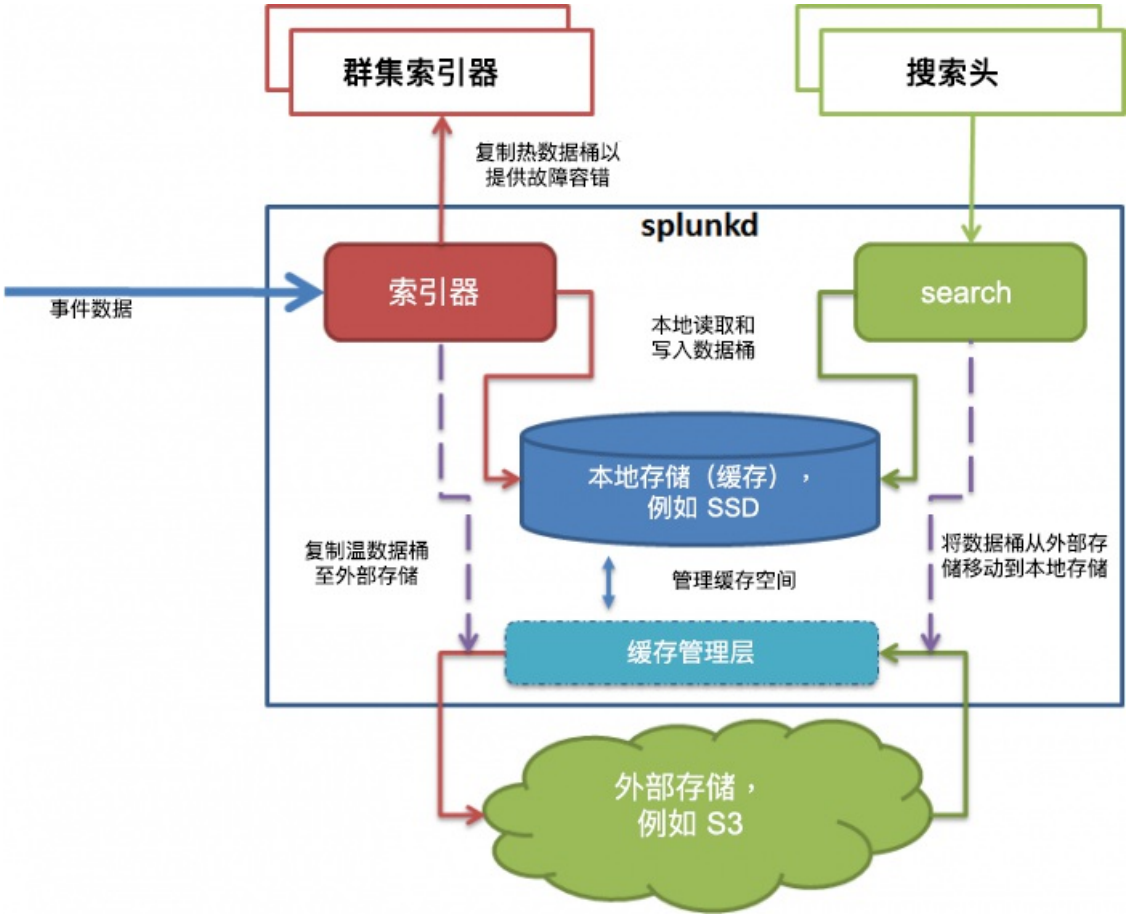
对于 SmartStore 索引，索引器群集仅保留热数据桶的复制和搜索因子副本。远程存储负责确保温数据桶的高可用性、数据保真度和灾难恢复。

因为远程存储会处理温数据桶高可用性，对等节点仅复制温数据桶的元数据而不是数据桶本身。这表示 SmartStore 索引的任何必要的数据桶修复都比非 SmartStore 索引快得多。

如果等于或超出复制因子的对等节点组减少，群集不会丢失任何 SmartStore 温数据桶数据，因为所有的温数据桶副本都会驻留在远程存储中。

SmartStore 数据流

此图表封装了索引器群集中启用 SmartStore 索引的数据流。虽然它包含对索引器群集的特定引用，但非群集索引器的重要架构是相同的。



数据传入来源索引器，索引器将索引数据并本地保存在热数据桶中。索引器还会将热数据桶数据复制到目标索引器中。到目前为止，数据流和非 SmartStore 索引的数据流相同。

但是，当热数据桶滚动成温数据桶之后，数据流会分散。来源索引器将温数据桶复制到远程对象存储中，同时将现有副本保留在缓存中，因为搜索往往会在最近的索引数据中运行。但是，目标索引器会删除副本，因为远程存储会确保高可用性，不需要保留多个本地副本。数据桶的主副本现驻留在远程存储中。

索引器上的缓存管理器是 SmartStore 数据流的中心。必要时缓存管理器会提取远程存储中的数据桶副本，以处理搜索请求。还会从缓存中逐出较旧或较少搜索的数据桶副本，因为它们参与搜索的可能性会随着时间的推移而降低。缓存管理器的任务是优化使用可用缓存，同时确保搜索可以立即访问需要的数据桶。

有关更多信息

请参阅章节“SmartStore 如何工作”深入了解 SmartStore 架构。其中包括以下主题：

- SmartStore 缓存管理器
- 索引如何在 SmartStore 中工作
- 搜索如何在 SmartStore 中工作
- 索引器群集操作和 SmartStore

部署 SmartStore

SmartStore 系统要求

启用 SmartStore 的索引器的要求和启用非 SmartStore 的索引器的要求基本相同。主要区别在于：

- 本地存储要求
- 需要连接远程存储

索引器要求

索引器硬件需求

硬件要求，除本地存储（详见下文）外，和其他 Splunk Enterprise 索引器相同。有关更多信息，请参阅：

- 本手册中的索引器群集的系统要求和其他部署注意事项
- *《容量规划手册》*中的“参考硬件”

SmartStore 操作一般对 CPU 性能没有明显影响。

索引器托管选项

用于 SmartStore 的对象存储类型决定了必须在哪里托管索引器：

- 运行带有 Amazon S3 远程存储的 SmartStore 的索引器必须托管在 AWS 上。
- 运行带有 GCP GCS 远程存储的 SmartStore 的索引器必须托管在 GCP 上。
- 运行带有兼容 S3 API 的本地对象存储的 SmartStore 的索引器必须托管在本地数据中心。

本地存储要求

根据您的索引器部署类型，按如下方式配置本地存储：

- 如果您是在本地 Linux 计算机上运行 Splunk Enterprise 索引器，首选的本地存储类型是 SSD。
- 如果您是在 AWS 上运行 Splunk Enterprise 索引器，请结合使用 AWS 和 SSD（例如 AWS I3、AWS R5）。
- 如果您是在 GCP 上运行 Splunk Enterprise 索引器，请针对分区 SSD 永久磁盘使用 N1 高内存计算机类型（n1-高内存-16 或 n1-高内存-32）。

每个索引器上可用于缓存数据的本地存储量必须和预期的工作集成比例。为获得最佳效果，预配能够容纳 30 天索引数据量的本地存储空间。例如，如果索引器每天添加约 100GB 的索引数据，则建议为缓存数据保留的存储空间大小为 3000GB。至少预配充足的存储空间以在缓存中保留至少 7 到 10 天的数据，因为通常会搜索过去 7 到 10 天索引的数据。

为和 Splunk Enterprise Security 一起使用，准备充分的本地存储以容纳 90 天的索引数据，而不是建议的 30 天。

注意：索引期间数据会压缩，因此索引数据的大小通常约为引入数据大小的 50%。但是，也有很多其他因素会影响索引数据和引入数据的大小比例。关于 Splunk Enterprise 数据量以及如何评估存储需求的常规讨论，请参阅*《安装手册》*中的“评估存储要求”。

关于索引器群集，决定存储要求的其他因素包括：

- 跨索引器群集分布引入数据。尽量在索引器之间平衡引入数据，这样所有的索引器能够索引、存储大致相当的数据量。
- 索引器中的热数据桶数量。热数据桶遵循复制和搜索因子策略，因此，每个数据桶上热数据桶的存储要求要比大小相同的温数据桶的要求更高。

要配置大小，缓存应与缓存所在分区相同。通常来说，该分区也包括很多其他文件，如 Splunk 二进制文件、操作系统、搜索项目、非 SmartStore 索引（如有）等。索引数据本身驻留在 $SPLUNK_DB 中。因此，预配存储和配置缓存大小时，您必须考虑此类文件。有关缓存大小的更多信息，请参阅“根据缓存磁盘分区的占用情况启动逐出”。

操作系统要求

如果您是在数据中心中运行 Splunk Enterprise，则每个本地计算机必须运行相同的 Linux 3.x + 64 位操作系统。

SmartStore 的本地部署不支持其他操作系统。

Splunk Enterprise 版本要求

独立索引器必须运行 Splunk Enterprise 7.2 或更高版本。

如果是索引器群集，群集中的所有节点（管理器节点、对等节点和搜索头）必须运行 Splunk Enterprise 7.2 或更高版本。其他版本兼容性要求适用于任何索引器群集中的节点，如“Splunk Enterprise 版本兼容性”中所述。

其他索引器群集要求

所有索引器群集要求适用于 SmartStore 索引。例如：

- 所有索引相关的设置，包括 SmartStore 设置，必须以相同方式跨对等节点配置。
- 您必须为所有对等节点上的同一组索引启用 SmartStore。
- 添加新索引段落时，必须将 `repFactor` 设置为 "auto"。

对于一般的索引器群集要求，请参阅“索引器群集的系统要求和其他部署注意事项”。

网络要求

索引器是远程存储的客户端，使用标准的 `https` 端口与之通信。

为了获得最佳性能，请针对从每个索引器到远程存储的连接使用 10Gbps 的网络连接。

远程存储要求

SmartStore 可以在 AWS S3 远程存储（包括 S3 API 兼容的本地对象存储）或 GCP GCS 远程存储上运行。根据您计划在哪种远程存储类型上部署，请参阅“为 SmartStore 配置 S3 远程存储”或“为 SmartStore 配置 GCS 远程存储”。

为 SmartStore 配置 S3 远程存储

在索引器上配置 SmartStore 设置之前，您必须确保远程存储设置无误，这样才可在索引器上使用。

之后，当您配置 SmartStore 的远程卷时，您可配置 `indexes.conf` 中远程存储的特定设置。索引器可使用这些设置和远程存储通信。

支持的远程存储服务

支持的远程存储服务包括 S3 和 Google GCS。有关 GCS 的信息，请参阅“为 SmartStore 配置 GCS 远程存储”。

对于 S3，SmartStore 可以使用：

- AWS S3
- 兼容 S3-API 的对象存储

要确定对象存储是否是 S3 兼容，请使用 S3 兼容性检查工具，位于：https://github.com/splunk/s3-tests。要使用工具，请遵循存储库的 README 文件中的说明使用。

配置 S3 远程存储

配置 S3 数据桶时：

- 数据桶必须具有读取、写入和删除权限。
- 如果正在 EC2 上运行索引器，请为数据桶配置 EC2 实例使用的相同区域。

请参阅 Amazon S3 文档获取如何创建和配置数据桶的信息。

对于通过 Splunk Enterprise 提供的特定于 S3 的设置，请搜索 indexes.conf 规格文件中以 `remote.s3` 开头的设置。

有关安全相关的设置的信息，如 S3 验证和加密设置，请参阅“使用 S3 时的 SmartStore 安全策略”。

容纳远程存储寻址模型

Amazon S3 目前支持两种寻址模型，或请求 URI 样式：路径样式，或 V1，和虚拟主机样式，或 V2。在 V1 中，数据桶名称位于 URI 路径中，例如 `//s3.amazonaws.com/<bucketname>/key`。在 V2 中，数据桶名称是域名的一部分，例如 `//<bucketname>.s3.amazonaws.com/key`。

Amazon 将不再支持 V1，并且以后将需要请求新的 S3 数据桶才能使用 V2 模型。

Splunk Enterprise 远程存储本机 S3 的寻址

Splunk Enterprise 会自动为 Amazon S3 数据桶容纳 V1 和 V2 模型。您可以使用其中任一种模型，但是，和 S3 通信时，Splunk Enterprise 会将 V1 URI 转换为 V2。

使用 V1 模型时，您可以指定 URI，如下所示：

```
[volume:s3volume]
storageType = remote
path = s3://<bucketname>/rest/of/path
```

使用 V2 模型时，您可以指定 URI，如下所示：

```
[volume:s3volume]
storageType = remote
path = s3://rest/of/path
remote.s3.bucket_name = <bucketname>
```

同样地，如果您指定以 amazonaws.com 结尾的端点，Splunk Enterprise 会根据端点确定 URI 版本，因为结构是固定的。例如：

```
[volume:s3volume]
storageType = remote
path = s3://<bucketname>/rest/of/path
remote.s3.endpoint = https://s3.us-west-1.amazonaws.com
```

或

```
[volume:s3volume]
storageType = remote
path = s3://rest/of/path
remote.s3.endpoint = https://<bucketname>.s3.us-west-1.amazonaws.com
```

两者指定相同的数据桶，Splunk Enterprise 会正确的解析其中一个。

Splunk Enterprise 远程存储与 S3 兼容的远程存储的寻址

如果您使用 S3 兼容的远程存储，而不是本机 S3，您可能需要指定 S3 兼容的存储支持的寻址模型。如果 S3 兼容的远程存储不支持 V1，您只需指定模型。

Splunk Enterprise 会提供 remote.s3.url_version 设置指定您正在使用的模型。其默认值为 v1。要将寻址模型更改为 V2，将设置更改为 v2。例如：

```
[volume:s3volume]
storageType = remote
path = s3://rest/of/path
remote.s3.url_version = v2
remote.s3.endpoint = https://bucketname.whatever.customer.com
```

为 SmartStore 配置 GCS 远程存储

在索引器上配置 SmartStore 设置之前，您必须确保远程存储设置无误，这样才可在索引器上使用。

之后，当您配置 SmartStore 的远程卷时，您可配置 indexes.conf 中远程存储的特定设置。索引器可使用这些设置和远程存储通信。

支持的远程存储服务

支持的远程存储服务包括 GCS 和 AWS S3。有关 S3 的信息，请参阅“为 SmartStore 配置 S3 远程存储”。

配置 GCS 远程存储

配置 GCS 数据桶时：

- 对于与索引器相关联的 GCS 存储桶，索引器的计算引擎服务帐户必须拥有读取、写入和删除权限。
- 配置数据桶，使其与索引器计算引擎实例在同一 GCP 区域中运行。

请参阅 Google GCS文档获取如何创建和配置数据桶的信息。

对于通过 Splunk Enterprise 提供的特定于 GCS 的设置，请搜索 indexes.conf 规格文件中以 remote.gs 开头的设置。

有关安全相关的设置的信息，如 GCS 验证和加密设置，请参阅“使用 GCS 时的 SmartStore 安全策略”。

为每个索引选择存储位置

您可以使用所有索引的相同设置全局配置 SmartStore，也可以基于每个索引配置 SmartStore。如果您在每个索引上配置 SmartStore，您可以在相同的索引器上或索引器群集上将 SmartStore 和非 SmartStore 索引混合在一起。您还可为不同的 SmartStore 索引指定不同的远程卷。

远程卷（如 indexes.conf 中定义）指向 SmartStore 存储温数据桶的远程存储，如 S3 或 GCS 数据桶上的位置。

要进行汇总，您可从以下存储选项中选择：

- 所有索引都远程存储在单个卷上。
- 所有索引都远程存储在多个卷上。
- 某些索引存储在本地，其他索引远程存储在一个或多个远程卷上。

即使使用 SmartStore 索引，一些索引数据也会临时存储在本地缓存中。但是，除了热数据桶，索引的数据桶主节点也是远程存储。为了简化操作，存储选项列表会在讨论存储卷时对索引的主节点副本做出假设，不会考虑 SmartStore 缓存中本地存储的数据。

请牢记以下限制：

- 每个远程卷限于单个索引器群集或独立索引器。也就是说，每个远程存储仅保留单个群集或独立索引器的数据桶。从配置的立场来看，indexes.conf 中各远程卷段落的 path 设置对群集或索引器来说必须唯一。例如，如果一个群集上的索引使用特定的远程卷，则任何其他群集或独立索引器上的索引都不能使用相同的远程卷。
- 每个 SmartStore 索引都限制为单个远程卷。该索引的所有温数据桶必须驻留在相同的远程存储中。
- 每个索引器或索引器群集都限制为跨所有索引的单个远程存储类型，如 S3 或 GCS。
- 索引器群集上的所有对等节点必须使用相同的 SmartStore 设置。

使用 S3 时的 SmartStore 安全策略

远程存储服务类型不同，SmartStore 安全策略也会有所不同。本主题介绍使用亚马逊简易存储服务 (S3) 作为远程存储服务时的安全策略。

使用远程存储服务进行验证

如何验证远程存储服务取决于索引器或索引器群集使用的云基础架构。

- 如果是在亚马逊弹性云计算 (EC2) 上运行索引器或索引器群集，则可以使用其身份和访问管理 (IAM) 角色中的访问密钥和密钥进行身份验证。
- 如果不是在 EC2 上运行索引器或索引器群集，请使用 indexes.conf 中的硬编码密钥。S3 硬编码密钥的相关设置如下：

- remote.s3.access_key：使用远程存储系统进行验证时要使用的访问密钥。
- remote.s3.secret_key：使用远程存储系统进行验证时要使用的密钥。
- remote.s3.endpoint：远程存储系统的 URL。此设置会告诉索引器进行 S3 验证的位置。使用 S3 数据桶区域的值。例如，https://s3.us-west-2.amazonaws.com。

有关这些设置的更多信息，请参阅 indexes.conf 规格文件。

您使用的凭据，无论是来自 IAM 角色还是 indexes.conf，都需要相关权限才能执行 S3 操作。如果您要加密远程存储上的静态数据，还需要相关权限才能执行亚马逊密钥管理服务 (KMS) 操作。

管理远程存储的 SSL 证书

SSL 证书设置因远程存储服务类型而有所差异。本部分提供有关使用 indexes.conf 中提供的设置管理 S3 远程存储的 SSL 的信息。有关这些设置的详细信息以及其他 S3 相关的 SSL 设置的信息，请参阅 indexes.conf 规范文件。

为远程卷配置 SSL 设置时，必须按卷逐一进行配置。这意味着您必须为在 indexes.conf 文件中定义的每个单独的远程卷指定 SSL 设置。

S3 SSL 设置叠加在 server.conf 配置文件的 sslConfig 段落中，但以下情况除外：

- sslVerifyServerCert
- sslAltNameToCheck
- sslCommonNameToCheck

如果在为 S3 存储配置 SSL 时遇到问题，请参阅 server.conf SSL 设置以及远程存储的特定设置。

下表包括常用设置以及建议值。

SSL 设置	描述	推荐的值
remote.s3.sslVerifyServerCert	指定是否检查 S3 端点提供的服务器证书。	true
remote.s3.sslVersions	要使用的 SSL 版本。	tls1.2
remote.s3.sslAltNameToCheck	要匹配的服务器中呈现的证书备用名称列表。例如，s3.<region>.amazonaws.com。	N/A
remote.s3.sslRootCAPath	含根证书列表的隐私增强邮件 (PEM) 格式文件的绝对路径。	N/A
remote.s3.cipherSuite	用于连接 S3 的密码。	与安全专家核实。以下为要为此设置输入的值类型示例： ECDHE-ECDSA-AES128-SHA256: ECDHE-ECDSA-AES256-SHA384: ECDHE-ECDSA-AES128-GCM-SHA256: ECDHE-ECDSA-AES256-GCM-SHA384: ECDHE-RSA-AES128-SHA:ECDHE-RSA-AES256-SHA: ECDHE-ECDSA-AES128-SHA: ECDHE-ECDSA-AES256-SHA: AES128-SHA256:AES256-SHA256: AES128-SHA:AES256-SHA: DHE-RSA-AES128-SHA: DHE-RSA-AES256-SHA: DHE-RSA-AES128-SHA256: DHE-RSA-AES256-SHA256
remote.s3.ecdhCurves	要发送的 Elliptic Curve-Diffie Hellman (ECDH) 曲线。	与安全专家核实。此处为要为此属性输入的值类型示例： prime256v1、secp384r1、secp521r1

对远程存储上的数据进行加密

SmartStore 支持在 S3 上对未使用的数据或既未使用也未在传输中的数据进行客户端和服务器端加密。SmartStore 通过 indexes.conf 配置文件中的 remote.s3.encryption 设置支持四种加密方案。您必须使用配置文件来加密 S3 卷上的数据，因为没有其他方法可以执行这种配置。

启用加密后，Splunk 平台会根据您使用的加密方法来生成加密密钥，并使用此密钥对您上传到目标卷的数据进行加密。在一定间隔之后，平台会生成一个新密钥。

对于大多数加密方法来说，执行加密的 Splunk 平台实例必须维持与 AWS 或 KMS 的连接。未能维持此连接可能会导致问题，即会生成用于加密的新密钥。

随后会启动对单个 Splunk 平台实例上的 SmartStore 卷的数据进行加密的高级过程。如果要加密的是索引器群集上 SmartStore 卷中的数据，请不要执行此过程，并转而参阅“更新通用对等节点配置和应用”。执行该过程时，请提供此过程中显示的设置。

1. 选择要在 SmartStore 卷上使用的加密方法。

   选择加密方法后不可更改。您之后无法再更改加密方法。

2. 在要对 SmartStore 卷中的数据进行加密的 Splunk 平台实例上，打开 $SPLUNK_HOME/etc/system/local/indexes.conf 进行编辑。
3. 通过使用该卷的 [volume:<volume_name>] 段落下的 remote.s3.encryption 设置，指定要应用于每个 SmartStore 卷的加密方法的类型：

   ```
   [volume:myvolume]
   remote.s3.encryption = sse-s3 | sse-kms | sse-c | cse | none
   ```
4. 根据您使用的加密类型，指定所需的其他设置以便通过与 AWS 或 KMS 交互来完成加密。若要了解要使用的设置，请参阅本主题后面提供的加密示例。

5. 保存 indexes.conf 文件并将其关闭，
6. 重新启动 Splunk 平台。

当数据上传到您指定的卷时会进行加密，并且加密操作会一直持续，直到您再次更改加密方案为止。为远程卷配置加密时，不会导致卷上已存在的数据被加密。

如果您禁用加密，不会导致现有的已加密数据被解密。Splunk 平台无法解密已加密数据，因此所有此类数据将不可用。

如果您尚不知道要使用哪种加密方案，则服务器端加密的最佳选择是 sse-c（使用客户密钥的服务器端加密）。此方法可避免遇到可能的 KMS 限制问题。对于客户端加密，cse 是最佳且唯一的选择。

- 有关用于加密的设置的详细信息，请参阅 indexes.conf 规格文件。
- 有关在 AWS 中配置服务器端加密的信息，请参阅 Amazon 文档。

使用客户提供的加密密钥 (sse-c) 进行服务器端加密

以下为用客户密钥设置服务器端加密的示例。所有这些设置都会进入 indexes.conf 配置文件。

```
[volume:example_volume]
remote.s3.encryption = sse-c
remote.s3.encryption.sse-c.key_type = kms
remote.s3.encryption.sse-c.key_refresh_interval = 86400
// 86400 equals 24 hours. This is the default and recommended value. The minimum value is 3600.
// Setting a very low value can degrade performance.
remote.s3.kms.auth_region = <aws_region>
remote.s3.kms.key_id = <kms_keyid>
// The kms_keyid must be a unique key ID, the Amazon Resource Name (ARN) of the CMK,
// or the name or ARN of an alias that points to the CMK.

// SSL settings for KMS communication
remote.s3.kms.sslVerifyServerCert = true
remote.s3.kms.sslVersions = tls1.2
remote.s3.kms.sslAltNameToCheck = kms.<aws_region>.amazonaws.com
remote.s3.kms.sslRootCAPath = $SPLUNK_HOME/etc/auth/kms_rootcert.pem
remote.s3.kms.cipherSuite = ECDHE-ECDSA-AES128-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-
AES256-GCM -SHA384:ECDHE-RSA-AES128-SHA:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-ECDSA-AES256-SHA:AES128 -
SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:DHE-RSA-AES128-SHA:DHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256
remote.s3.kms.ecdhCurves = prime256v1, secp384r1, secp521r1
```

使用 Amazon S3 管理的加密密钥 (sse-s3) 进行服务器端加密

以下为用 AES256 设置服务器端加密的示例。所有这些设置都会进入 indexes.conf 配置文件。

```
[volume:example_volume]
remote.s3.encryption = sse-s3
```

使用存储在 AWS KMS 中的客户主密钥 (sse-kms) 进行服务器端加密

以下为用 KMS 托管的密钥设置服务器端加密的示例。所有这些设置都会进入 indexes.conf 配置文件。

```
[volume:example_volume]
remote.s3.encryption = sse-kms
remote.s3.kms.key_id = <kms_keyid>
```

客户端加密 (cse)

客户端数据加密可确保云服务提供商 (CSP) 无法以任何方式读取加密的数据。

SmartStore 按您指定的时间间隔联系 AWS KMS 服务。它使用 KMS，根据 KMS 存储的客户主密钥 (CMK) 生成数据加密密钥 (DEK)。当您在启用 CSE 后上载数据时，SmartStore 会将数据加密密钥与上载的数据桶一起存储。

该软件不支持任何形式的密钥撤销。您负责管理客户主密钥和所有 DEK。同样，您不能加密已经用新 DEK 加密的数据。您必须先解密数据，然后让平台生成新的 DEK，然后再重新上传数据。

您必须使用 AWS KMS 才能利用此功能。它不适用于任何其他类型的密钥管理服务。

在 SmartStore 中的索引上启用客户端加密有一些注意事项：

- 数据加密可能使性能下降最高达 20%。
- 出于维护或故障排除的目的，您可以在 SmartStore 卷上运行的某些远程文件系统 CLI 命令有些限制：
  - `splunk cmd splunkd rfs getF` 命令只会让您上载 `receipt.json` 文件，而且 CSE 不会在上载时对该文件进行加密。
  - `splunk cmd splunkd rfs putF` 命令只会让您下载 `receipt.json` 文件，并假定该文件尚未加密。
  - `splunk cmd splunkd rfs get` 命令必须包含 `receipt.json`（作为参数），否则命令会失败。

以下为设置客户端加密的示例。所有这些设置都会进入 `indexes.conf` 配置文件。

```
# The volume stanza and path specify the name of the remote volume.
# This is the volume that the index that will store the encrypted data uses.
# The Splunk platform expects the path to be in the remote storage format.
[volume:<VOLUME_NAME>]
path = <S3_BUCKET_PATH>
storageType = remote

# The following settings facilitate interaction with AWS KMS. You need
# KMS to generate the client-side encryption key. The settings
# set the the KMS endpoint and authentication region,
#
# You must configure one of the following two settings
# If you configure neither setting, the Splunk platform attempts
# to use the 'remote.s3.endpoint' and 'remote.s3.auth_region' settings
# before it fails to start.
remote.s3.kms.endpoint = <KMS_ENDPOINT>
remote.s3.kms.auth_region = <KMS_AUTH_REGION>

# The unique ID or Amazon Resource Name (ARN) of the
# customer master key to use, or the alias name or ARN
# of the alias that refers to this key.
remote.s3.kms.key_id = <KEY_ID>

# The signature version to use when authenticating the remote storage
# system supporting the S3 API. Since CSE uses KMS to manage encryption
# keys, you must set this to "v4".
remote.s3.signature_version = v4

# Enable client-side encryption on the remote volume.
remote.s3.encryption = cse

# The bucket encryption algorithm to use for CSE.
# Currently, "aes-256-gcm" is the only acceptable value.
remote.s3.encryption.cse.algorithm = aes-256-gcm

# How long to wait, in seconds, between generation of
# keys to encrypt data that is uploaded to S3 when CSE
# is active.
remote.s3.encryption.cse.key_refresh_interval = 86400

# The key mechanism to use for CSE. Currently, "kms"
# is the only acceptable value. You must configure this
# setting for the Splunk platform to start.
remote.s3.encryption.cse.key_type = kms


[INDEX_NAME]
# The index you specify for CSE of data. When the Splunk
# platform adds data to this index, it encrypts the data with
# the key that you provide. The index references
# the remote S3 storage volume.
homePath = $SPLUNK_DB/<INDEX_NAME>/db
coldPath = $SPLUNK_DB/<INDEX_NAME>/colddb
thawedPath = $SPLUNK_DB/<INDEX_NAME>/thaweddb
remotePath = volume:<VOLUME_NAME>/$_index_name
```

使用 GCS 时的 SmartStore 安全策略

远程存储服务类型不同，SmartStore 安全策略也会有所不同。本主题介绍使用 Google 云存储（GCS）作为远程存储服务时的安全策略。

使用远程存储服务进行验证

索引器使用凭据文件对 GCP 进行验证。

索引器如何获得凭据

索引器可以通过多种方式获得凭据。按优先级从高到低的顺序，索引器会使用：

1. 由 indexes.conf 中的 remote.gs.credential_file 设置（如果有）指定的凭据文件。
2. 与 indexes.conf 中的 remote.gs.service_account_email 设置（如果有）相关联的帐户的凭据，
3. 计算引擎的默认服务帐户的凭据。

在索引器上指定自定义凭据

要在索引器上指定自定义凭据，请使用 indexes.conf 中的 remote.gs.credential_file 设置：

```
remote.gs.credential_file = <credentials.json>
```

此设置指定的文件必须位于索引器上，如下所示：

- 对于独立索引器，文件必须位于 $SPLUNK_HOME/etc/auth 下。
- 对于群集中的对等节点，您可以通过配置软件包的方法分发文件，将文件放在管理器节点上的 $SPLUNK_HOME/etc/master-apps/_cluster/local 目录下，然后将软件包分发给每个对等节点，使文件驻留在其 $SPLUNK_HOME/etc/slave-apps/_cluster/local 目录下。您还可以将文件放在每个对等节点上的 $SPLUNK_HOME/etc/auth 中。分布式软件包位置优先。

凭据文件在启动时进行加密。

凭据权限

您使用的凭据需要权限才能执行 GCS 操作。如果您使用的是 gcp-sse-c 或 gcp-sse-kms 加密方案，他们还需要获得执行 GCP 密钥管理服务（KMS）操作的权限。

管理远程存储的 SSL 证书

SSL 证书设置因远程存储服务类型而有所差异。本部分提供有关使用 indexes.conf 中提供的设置管理 GCS 远程存储的 SSL 的信息。有关这些设置的详细信息以及其他 GCS 相关的 SSL 设置的信息，请参阅 indexes.conf 规格文件。

GCS SSL 设置叠加在 cipherSuite 和 caCertFile（sslRootCAPath）的 server.conf 的 sslConfig 段落中。因此，如果遇到问题，除了远程存储专有的设置外，还请查阅 server.conf 设置。

根据每个远程卷指定 SSL 设置。

下表包括常用属性以及建议值。

SSL 设置	描述	推荐的值
remote.gs.sslVerifyServerCert	指定是否检查 GCS 端点提供的服务器证书。	true
remote.gs.sslVerifyServerName	指定是否验证服务器凭据中的“常用名”或“主题备用名”是否与它连接的 URL 中的主机名匹配。	true
remote.gs.sslVersionsForClient	指定用于传出连接的最低 SSL/TLS 版本。	tls1.2
	包含根证书列表的	

remote.gs.sslRootCAPath	PEM 格式文件的绝对路径。	N/A
remote.gs.cipherSuite	用于连接 GCS 的密码。	与安全专家核实。此处为要为此属性输入的值类型示例： ECDHE-ECDSA-AES128-SHA256:ECDHE-ECDSA-AES256-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-SHA:ECDHE-RSA-AES256-SHA:ECDHE-ECDSA-AES128-SHA:ECDHE-ECDSA-AES256-SHA:AES128-SHA256:AES256-SHA256:AES128-SHA:AES256-SHA:DHE-RSA-AES128-SHA:DHE-RSA-AES256-SHA:DHE-RSA-AES128-SHA256:DHE-RSA-AES256-SHA256

对远程存储上的数据进行加密

SmartStore 支持对GCS 上的静态数据进行服务器端加密。SmartStore 支持使用 indexes.conf 中的 remote.gs.encryption 属性的三种加密方案：

```
remote.gs.encryption = gcp-sse-c | gcp-sse-kms | gcp-sse-gcp
```

默认值为 "gcp-sse-gcp"。

基于卷配置属性。

推荐的远程存储加密方法是 "gcp-sse-c"（使用客户密钥进行服务器端加密）。此方法可避免遇到可能的 KMS 限制问题。

选择加密方法后不可更改。您之后无法再更改加密方法。

有关加密模式的详细信息，请参阅：

- index.conf 规格文件，了解 remote.gs 加密相关的设置。
- Google 文档，了解有关配置云存储加密密钥的信息。

用 gcp-sse-c 加密

以下为用客户提供的加密密钥设置服务器端加密的示例：

```
[volume:my_gcs_vol]
storageType = remote
path = gs://your_gcp_test_1
remote.gs.project_id = your-app-test
remote.gs.encryption = gcp-sse-c
remote.gs.encryption.gcp-sse-c.key_refresh_interval = 86400
remote.gs.encryption.gcp-sse-c.key_type = gcp_kms
remote.gs.gcp_kms.locations = us-central1
remote.gs.gcp_kms.key_ring = your_ring_1
remote.gs.gcp_kms.key = test_key_1
```

用 gcp-sse-gcp 加密

以下为用 Google 管理的加密密钥设置服务器端加密的示例：

```
[volume:my_gcs_vol_3]
storageType = remote
remote.gs.encryption = gcp-sse-gcp
path = gs://your_gcp_test_3
```

用 gcp-sse-kms 加密

在配置 GCS 数据桶时，您可以指定密钥，也可以按默认方式使用与 GCS 数据桶关联的密钥。

以下为用客户管理的加密密钥设置服务器端加密的示例：

```
[volume:my_gcs_vol_2]
storageType = remote
path = gs://your_gcp_test_2
```

```
remote.gs.project_id = your-app-test
remote.gs.encryption = gcp-sse-kms
remote.gs.gcp_kms.locations = us-central1
remote.gs.gcp_kms.key_ring = your_ring_us
remote.gs.gcp_kms.key = test_key_3
```

如果要使用与 GCS 数据桶服务帐户关联的密钥，请将密钥设置保留为空：

```
[volume:my_gcs_vol_2]
storageType = remote
path = gs://your_gcp_test_2
remote.gs.project_id = your-app-test
remote.gs.encryption = gcp-sse-kms
```

在新的索引器群集上部署 SmartStore

在此过程中，您可：

1. 安装新的索引器群集。
2. 配置群集对等节点以访问远程存储。
3. 测试部署。

注意：此流程仅为新的索引器群集配置 SmartStore。不介绍如何在索引器群集上上载现有数据。要将现有的本地索引迁移到 SmartStore 中，请参阅“将索引器群集上的现有数据迁移到 SmartStore 中”。要在新的索引器群集上启用现有的 SmartStore 数据，请参阅“启动 SmartStore 索引”。

要在独立索引器上部署 SmartStore，请参阅“在新的独立索引器上部署 SmartStore”。

前提条件

阅读：

- 索引器群集部署概述
- 更新通用对等节点配置和应用
- SmartStore 系统要求
- 为 SmartStore 配置 S3 远程存储，或为 SmartStore 配置 GCS 远程存储
- 为每个索引选择存储位置
- 您正在使用的远程存储服务提供商提供的文档

警告

小心这些配置问题：

- 各远程卷段落的 `path` 设置值对于索引器群集来说必须唯一。您只能在单个群集中的各索引之间共享远程卷。换句话说，如果一个群集上的索引使用特定的远程卷，则任何其他群集或独立索引器上的索引都不能使用相同的远程卷。
- 您必须在索引器群集中设置所有 SmartStore 索引才能使用 `repfactor = auto`。
- 将 `maxDataSize` 保留为每个 SmartStore 索引的默认值 "auto"（750MB）。
- 每个 SmartStore 索引的 `coldPath` 设置需要一个值，即使在迁移索引的情况下，也会忽略该设置。
- 每个 SmartStore 索引的 `thawedPath` 设置需要一个值，即使该设置没有实际用途，因为您无法将数据解冻为 SmartStore 索引。请参阅“解冻数据和 SmartStore”。

部署 SmartStore

以下流程假设您正在新的索引器群集上部署 SmartStore。还假设您正在使用单个远程位置为群集上的所有索引部署 SmartStore。如果您只想为某些索引部署 SmartStore，或者您为 SmartStore 索引使用多个远程位置，您可修改流程以满足您的需求。

1. 确保您满足前提条件。尤其是，请阅读以下内容：
   - SmartStore 系统要求
   - 为 SmartStore 配置 S3 远程存储，或为 SmartStore 配置 GCS 远程存储。
2. 理解 SmartStore 安全策略并在必要时在部署进程中准备实施。请参阅“采用 S3 安全策略的 SmartStore”或“采用 GCS 安全策略的 SmartStore”。
3. 安装新的 Splunk Enterprise 实例并将其作为新的索引器群集的管理器节点。请参阅“启用索引器群集管理器节点”。
4. 将索引器群集的复制因子和搜索因子设为相同值，如 3/3。
5. 在管理器节点上，创建或编辑 `$SPLUNK_HOME/etc/master-apps/_cluster/local/indexes.conf` 并指定 SmartStore 设置，如以下示例所示。当对等节点稍后启动时，管理器节点会自动将这些设置以及配置软件包的其余部分分发到对等节点。

使用 S3 远程对象存储：
此示例使用 S3 远程对象存储配置 SmartStore 索引。在全局级别配置 SmartStore 相关的设置，这样所有索引器都会启用 SmartStore，它们都使用一个名为 "remote_store" 的远程存储卷。示例还会新建一个名为 "cs_index" 的索引。

```
[default]
# Configure all indexes to use the SmartStore remote volume called
# "remote_store".
# Note: If you want only some of your indexes to use SmartStore,
# place this setting under the individual stanzas for each of the
# SmartStore indexes, rather than here.
remotePath = volume:remote_store/$_index_name

repFactor = auto

# Configure the remote volume.
[volume:remote_store]
storageType = remote

# The volume's 'path' setting points to the remote storage location where
# indexes reside. Each SmartStore index resides directly below the location
# specified by the 'path' setting.
path = s3://mybucket/some/path

# The following S3 settings are required only if you're using the access and secret
# keys. They are not needed if you are using AWS IAM roles.

remote.s3.access_key = <S3 access key>
remote.s3.secret_key = <S3 secret key>
remote.s3.endpoint = https:|http://<S3 host>

# This example stanza configures a custom index, "cs_index".
[cs_index]
homePath = $SPLUNK_DB/cs_index/db
# SmartStore-enabled indexes do not use thawedPath or coldPath, but you must still specify them here.
coldPath = $SPLUNK_DB/cs_index/colddb
thawedPath = $SPLUNK_DB/cs_index/thaweddb
```

关于这些设置的详细信息，请参阅“配置 SmartStore”。另请参阅*《管理员手册》*中的 indexes.conf.spec。

使用 GCS 远程对象存储：
此示例使用 GCS 远程对象存储配置 SmartStore 索引。在全局级别配置 SmartStore 相关的设置，这样所有索引器都会启用 SmartStore，它们都使用一个名为 "remote_store" 的远程存储卷。示例还会新建一个名为 "cs_index" 的索引。

```
[default]
# Configure all indexes to use the SmartStore remote volume called
# "remote_store".
# Note: If you want only some of your indexes to use SmartStore,
# place this setting under the individual stanzas for each of the
# SmartStore indexes, rather than here.
remotePath = volume:remote_store/$_index_name

repFactor = auto

# Configure the remote volume.
[volume:remote_store]
storageType = remote

# The volume's 'path' setting points to the remote storage location where
# indexes reside. Each SmartStore index resides directly below the location
# specified by the 'path' setting.
path = gs://mybucket/some/path

# There are several ways to specify credentials. For details, see the topic,
# "SmartStore on GCS security strategies." One way to specify credentials
# is to point to a file, as shown here.
remote.gs.credential_file = credential.json

# This example stanza configures a custom index, "cs_index".
```

```
[cs_index]
homePath = $SPLUNK_DB/cs_index/db
# SmartStore-enabled indexes do not use thawedPath or coldPath, but you must still specify them here.
coldPath = $SPLUNK_DB/cs_index/colddb
thawedPath = $SPLUNK_DB/cs_index/thaweddb
```

关于这些设置的详细信息，请参阅“配置 SmartStore”。另请参阅*《管理员手册》*中的 indexes.conf.spec。

6. 仅在管理器节点上，运行：

```
splunk apply cluster-bundle --answer-yes
```

7. 为所有新的索引器群集安装并启用对等节点和搜索头。请参阅“启用对等节点”和“启用搜索头”。稍微等待对等节点下载具有 SmartStore 设置的配置软件包。要查看配置软件包进程状态，您可运行 `splunk show cluster-bundle-status` 命令，详见“更新常见对等节点配置和应用”中的介绍。
8. 测试部署。
   1. 您可使用此命令通过管理器节点监视群集启动过程的状态：

      ```
      splunk show cluster-status -auth <admin>:<password>
      ```
   2. 要确认跨索引器群集的远程存储访问：
      1. 将样本文本文件放在远程存储中。
      2. 从一个对等节点中，运行此命令以递归方式列出了远程存储中显示的所有文件：

         ```
         splunk cmd splunkd rfs -- ls --starts-with volume:remote_store
         ```

      如果您运行命令时看到样本文件，则表示您具有远程存储访问权限。

   3. 验证数据是否传输到远程存储：
      1. 将一些数据发送到索引器。
      2. 等待要滚动的数据桶。如果您不想等数据桶自然滚动，您可通过对等节点手动滚动某些数据桶：

         ```
         splunk _internal call /data/indexes/<index_name>/roll-hot-buckets -auth <admin>:<password>
         ```
      3. 查找要上载到远程存储的温数据桶。
   4. 验证是否从远程存储传输数据：

      注意：此时，您应能够针对此数据运行正常搜索。大多数情况下，您不会传输远程存储中的任何数据，因为数据将保留在本地缓存中。因此，要验证是否从远程存储中传输数据，建议您首先将数据桶从其中一个对等节点的本地缓存中逐出。
      1. 在其中一个对等节点上，使用此 POST 将数据桶从缓存逐出到 REST 端点上：

         ```
         services/admin/cacheman/<cid>/evict
         ```

         其中 `<cid>` 是 `bid|<bucketId>|`。例如："bid|cs_index~0~7D76564B-AA17-488A-BAF2-5353EA0E9CE5|"

         注意：要获取数据桶的 `bucketId`，请转到搜索头节点，然后在测试索引上运行一个搜索。例如：

         ```
         splunk search "|rest /services/admin/cacheman | search title=*cs_index*  | fields splunk_server, title" -auth
         <admin>:<password>
         ```

         结果列出了指定测试索引中的数据桶集（按 `bucketId` 列出）以及它们的关联对等节点。您可以基于这些信息，决定从其中一个对等节点的缓存中逐出一个数据桶。

      2. 在对等节点本地运行搜索。搜索必须需要被逐出的数据桶中的数据。

      对等节点必须立即传输远程存储中的数据桶以运行搜索。运行搜索之后，您可检查数据桶是否重新显示在缓存中。

后续步骤

使用 SmartStore 运行群集之后，需要立刻关注一些配置问题。特别是：

- 在管理器节点上，编辑 `$SPLUNK_HOME/etc/master-apps/_cluster/local/indexes.conf` 文件，配置数据保留设置以确保群集符合您需要的冻结行为。请参阅“为 SmartStore 索引配置数据保留”。

这一步极其重要，可避免不需要的数据桶冻结和可能的数据丢失。SmartStore 数据桶冻结行为和设置和非 SmartStore 行为和设置不同。

- 在管理器节点上，编辑 `$SPLUNK_HOME/etc/master-apps/_cluster/local/server.conf` 以对对等节点上 SmartStore 相关的 `server.conf` 设置进行必要更改。尤其是，配置缓存大小以满足部署需求。请参阅“配置 SmartStore 缓存管理器”。

对管理器节点上的配置软件包做出这些更改后，应用软件包以将这些设置分发到对等节点：

```
splunk apply cluster-bundle --answer-yes
```

关于其他 SmartStore 设置的详细信息，请参阅“配置 SmartStore”。

使用 SmartStore 部署多站点索引器群集

您可以使用 SmartStore 部署多站点索引器群集，以满足灾难恢复要求。群集及其对象存储可以托管在公共云提供商或本地数据中心。

有以下 SmartStore 多站点部署类型可用：

- 在公共云提供商（AWS 或 GCP）中托管的双活多站点，位于跨多个小区域的单个大区域内
- 跨数据中心托管的双活多站点，使用符合 S3 API 的本地对象存储

多站点部署必须托管在单个公共云提供商或一组本地数据中心内。各站点不能跨越多个云提供商，也不能跨越云提供商和数据中心。

使用 SmartStore 部署多站点索引器群集的要求

使用 SmartStore 部署多站点索引器群集时，需要考虑几层要求：

- 所有多站点群集和所有启用 SmartStore 的群集的标准要求。请参阅“多站点群集和启用 SmartStore 的群集的标准要求”。
- 使用 SmartStore 部署的所有多站点群集的要求，与部署类型无关。请参阅“使用 SmartStore 部署的所有多站点群集的要求”。
- 每种部署类型的专有要求。
  - 对于所托管的公共云提供商，请参阅“所托管的单区域内公共云提供商”。
  - 有关本地托管，请参阅“跨数据中心的本地托管”。

多站点群集和启用 SmartStore 的群集的标准要求

多站点群集和启用 SmartStore 的群集的标准要求和部署方法也适用于使用 SmartStore 部署的多站点群集组合。请参阅本手册中的相关主题，尤其包括：

- 多站点索引器群集部署概述
- SmartStore 系统要求
- 在新的索引器群集上部署 SmartStore

在您完全了解多站点索引器群集和启用了 SmartStore 的索引器群集的要求和部署方法之前，请不要进行部署。

使用 SmartStore 部署的所有多站点群集的要求

除了这些标准要求之外，使用 SmartStore 部署的多站点群集还有其他一些要求。本节中列出的要求适用于使用 SmartStore 的所有多站点群集部署情况。

部署要求

这些要求适用于使用 SmartStore 的所有多站点群集部署情况：

- 指定一个站点为主站点。该站点托管活跃群集管理器节点。
- 每个群集站点必须有一个管理器节点。在任何时间点，群集中必须有且只有一个管理器节点处于活跃状态。其他管理器节点必须保持在备用模式。
- 每个站点上通常会部署搜索头，但这一点并非强制性要求。
- 站点位置以双活复制关系托管两个对象存储。根据部署类型，群集对等节点集可以将数据发送到一个对象存储或两个都发。

配置要求

这些配置对于所有使用 SmartStore 部署的多站点群集来说都必不可少：

- 在每个搜索头上，在 server.conf 中设置 `site=site0` 以禁用站点相关性。
- 在管理器节点上，在 server.conf 中配置 `site_replication_factor` 和 `site_search_factor` 以确保每个站点至少每个数据桶都有一个可搜索副本。
- 为确保在一个或多个对等节点发生故障时，群集仍能继续引入所有传入的数据，请配置转发器以在所有站点的所有对等节

点之间实现数据的负载均衡。请参阅“使用转发器将数据导入索引器群集”。

准备备用管理器节点

每个站点位置必须托管一个管理器节点。正常运行的群集只有一个活跃管理器节点。其他管理器节点必须保持在备用模式，以便在活跃管理器所在的站点发生故障时可以使用这些节点。

请参阅“处理管理器站点故障”，了解有关如何准备和实施管理器节点故障转移操作的详细信息。以下是本主题中介绍的主要准备步骤：

1. 在每个站点上设置一个管理器节点。
2. 将主站点上的管理器节点设置为活跃管理器。
3. 将活跃管理器的配置复制到每个备用管理器。
4. 持续将所有新配置或更改的配置从活跃管理器复制到备用管理器。
5. 制定计划，以便在活跃管理器发生故障时通过明确故障转移操作转至备用管理器。

所托管的单区域内公共云提供商

AWS 和 GCP 云提供商都可以使用这种部署类型。由 AWS 托管的部署将 S3 用作远程存储。由 GCP 托管的部署将 GCS 用作远程存储。

在这种部署类型中，每个索引器群集站点都托管在单个大区域内的单独小区域中。每个站点/小区域中的索引器均配置有相同的 AWS S3 或 GCS 端点。如果任何小区域发生故障（包括整个小区域故障），公共云提供商会针对对象存储在大区域内的各小区域之间执行故障转移。

此外，每个小区域都必须托管一个群集管理器节点。一次只能有一个管理器处于活跃状态。其他管理器必须准备好作为备用故障转移。

部署拓扑

以下是部署拓扑的基本元素：

- 每个索引器群集站点驻留在单个大区域内的单独小区域中。
- 小区域之间的网络延迟小于 15 毫秒。
- 一个典型部署总共包含三个站点（小区域）。
- 每个小区域都托管一个活跃或备用群集管理器节点。
- 所有站点上的所有对等节点都指向单个对象存储 URI 端点。
- 公共云提供商处理同一大区域中跨小区域的对象存储之间的同步复制，并在必要时执行故障转移。

站点故障和恢复

如果站点发生故障，请注意以下几点：

- 如果故障站点托管了活跃群集管理器，则必须立即切换到其余站点之一的备用管理器节点。请参阅“处理管理器站点故障”。
- 公共云提供商会自动处理小区域内的对象存储故障。
- 配置为在跨群集内确保负载平衡的转发器会自动重定向到其余站点上的对等站点。
- 如果搜索头正在访问发生故障的站点上的对等站点，则站点发生故障时正在进行的任何搜索都可能返回不完整的结果。发生故障后，新搜索将自动重定向到其余站点上的对等站点。

当故障站点返回群集时，您必须处理以下与群集管理器相关的问题：

1. 确保已恢复站点上的对等节点指向新的活跃管理器。
2. 由于已恢复站点的管理器现在充当备用管理器，因此，如“准备备用管理器节点”中所述，请将其包含在将来的配置更新中。

跨数据中心的本地托管

此部署类型仅适用于符合 S3-API 的对象存储。

此部署类型限于两个站点，而且每个站点都托管在本地数据中心。两个站点都托管了复制的活跃对象存储。一个站点托管一个活跃群集管理器，另一个站点则托管一个备用群集管理器。

部署拓扑

以下是部署拓扑的基本元素：

- 每个索引器群集站点都驻留在本地数据中心。
- 站点之间的网络延迟最长为 300 毫秒。为了获得最佳性能，建议的最大延迟为 100 毫秒。
- 此拓扑限于两个站点。
- 每个站点位置都托管一个活跃对象存储。
- 每个对等节点都通过第三方 VIP 或 GSLB 指向相同的远程对象存储 URI，
- VIP 或 GSLB 将流量从每个对等节点发送到位于其站点位置的对象存储。如果对象存储失败，VIP 或 GSLB 会根据需要将流量重新发送到剩余的活跃对象存储。
- 一个站点位置托管一个活跃群集管理器，另一个站点位置则托管一个备用群集管理器节点。

对象存储要求

有关 SmartStore 的 S3 远程对象存储的基本要求，请参阅“为 SmartStore 配置 S3 远程存储”。

对于本地多站点部署，第三方对象存储的要求包括以下内容：

- 必须符合 S3 API。
- 支持物理对象存储之间的双向复制。
- 重试将上传的对象复制到目标存储，以便在对象存储失败和恢复的情况下确保数据同步。
- 支持类似于 AWS S3 的对象版本控制。对象的版本控制必须基于对象创建或修改时间戳，而不是复制时间戳。
- 支持删除标记复制。

此外，对象存储供应商必须确保远程存储之间的最大复制延迟时间不超过您的恢复点目标（RPO）要求。

正常运行的详细信息

相较于启用了 SmartStore 的单站点群集中的对等节点，启用了 SmartStore 的多站点群集中的对等节点从远程存储中上载和下载数据没有什么不同。群集中所有对等节点的 path 设置必须使用相同的 URI 才能标识远程存储。区别在于，URI 指向将流量发送到对等节点的站点位置（数据中心）的活跃远程存储的 VIP 或 GSLB。因此，在正常运行时，对等节点始终会将数据上载到其本地远程存储并从其本地远程存储下载数据。

上载到一个远程存储的数据会使用远程存储供应商提供的功能复制到另一个远程存储。复制解决方案通常依赖于异步复制，这会导致延迟，从而导致复制到非本地存储的最新数据暂时不可用。复制延迟的量会因上载量、位置之间的可用网络带宽以及供应商专有的复制延迟的不同而有所不同。

复制延迟是异步复制的固有特征。Splunk Enterprise 对由于复制延迟而尚未复制到第二个对象存储的数据不提供任何其他保护。

由于禁用了站点相关性，因此搜索头可以访问任一站点上的对等节点，以满足搜索请求。这可能导致跨数据中心的 WAN 流量。

如果请求的数据在本地对象存储上尚不可用，但正在复制到本地对象存储中，则涉及不驻留在本地对等节点缓存上的数据的搜索请求可能返回不完整的数据。这种情况会影响复制延迟时间范围内的最新数据集。

站点故障和恢复

如果站点位置故障，请注意以下几点：

- 如果故障站点托管了活跃群集管理器，则必须立即切换到其余站点的备用管理器节点。请参阅“处理管理器站点故障”。
- 配置为在跨群集内确保负载平衡的转发器会自动重定向到其余站点上的对等站点。
- 如果搜索头正在访问发生故障的站点上的对等站点，则站点发生故障时正在进行的任何搜索都可能返回不完整的结果。发生故障后，新搜索将自动重定向到其余站点上的对等站点。
- 上载到其余对象存储的数据不会复制到故障位置上的对象存储。
- 由于复制延迟，其余对象存储可能缺少最近上载到故障对象存储的数据。这种情况是暂时的，当故障对象存储恢复服务时即会恢复正常。

如果发生灾难性故障而导致对象存储永久性失效，则最近上传到该对象存储中但未复制到其余对象存储中的数据可能会永久丢失。在这种情况下，针对已丢失数据桶的访问请求，群集会发出不完整搜索结果警告和数据桶修复错误。要消除这些警告和错误，请联系 Splunk 客户支持以获取有关删除相应数据桶元数据的帮助。

当故障站点返回群集时，您必须处理以下与群集管理器相关的问题：

1. 确保已恢复站点上的对等节点指向新的活跃管理器。
2. 由于已恢复站点的管理器现在充当备用管理器，因此，如“准备备用管理器节点”中所述，请将其包含在将来的配置更新中。

当发生故障的对象存储恢复正常时，对象存储复制会重新开始，对象存储还会复制故障期间未发送的所有数据，以及在另一个对象存储关闭时上载到其余对象存储的数据。重新同步对象存储所需的时间取决于多种因素，例如故障的持续时间（以及累积的未复制数据量）和复制吞吐量。

当对象存储重新同步时，搜索请求可能会产生不完整的结果。由于恢复的对象存储可能会丢失大量的最新数据，因此它需要一

些时间通过重新同步过程来跟上进度。若有用户尝试访问恢复对象存储上尚未同步到该存储的数据，则可能会产生部分搜索结果。

仅与对象存储相关的故障和恢复

如果对象存储发生故障，但其对应的群集站点仍处于活跃状态，则 VIP 或 GSLB 会将流量从站点的对等节点重定向到其余的对象存储。这会导致跨数据中心的 WAN 流量增加。

由于复制延迟，其余对象存储可能缺少最近上传到故障对象存储的数据，这可能会导致搜索结果不完整。这种情况是暂时的，当故障对象存储恢复服务时即会恢复正常。

如果发生灾难性故障而导致对象存储永久性失效，则最近上传到该对象存储中但未复制到其余对象存储中的数据可能会永久丢失。在这种情况下，针对已丢失数据桶的访问请求，群集会发出不完整搜索结果警告和数据桶修复错误。要消除这些警告和错误，请联系 Splunk 客户支持以获取有关删除相应数据桶元数据的帮助。

当发生故障的对象存储恢复正常时，对象存储复制会重新开始，对象存储还会复制故障期间未发送的所有数据，以及在另一个对象存储关闭时上载到其余对象存储的数据。重新同步对象存储所需的时间取决于多种因素，例如故障的持续时间（以及累积的未复制数据量）和复制吞吐量。

当对象存储重新同步时，搜索请求可能会产生不完整的结果。由于恢复的对象存储可能会丢失大量的最新数据，因此它需要一些时间通过重新同步过程来跟上进度。若有用户尝试访问恢复对象存储上尚未同步到该存储的数据，则可能会产生部分搜索结果。

在新的独立索引器上部署 SmartStore

在此过程中，您可：

1. 安装新的索引器。
2. 配置索引器以访问远程存储。
3. 测试部署。

注意：此流程仅为新的独立索引器配置 SmartStore。不介绍如何将现有数据上载至索引器。要将现有的本地索引迁移到 SmartStore 中，请参阅“将独立索引器上的现有数据迁移到 SmartStore 中”。要在新的独立索引器上启用现有的 SmartStore 数据，请参阅“启动 SmartStore 索引”。

要在群集上部署 SmartStore，请参阅“在新的索引器群集上部署 SmartStore”。

前提条件

阅读：

- SmartStore 系统要求
- 为 SmartStore 配置 S3 远程存储，或为 SmartStore 配置 GCS 远程存储。
- 为每个索引选择存储位置
- 您正在使用的远程存储服务提供商提供的文档

警告

小心这些配置问题：

- 各远程卷段落的 `path` 设置值对于索引器来说必须唯一。您只能在单个独立的索引器中的各索引之间共享远程卷。换句话说，如果一个索引器上的索引使用特定的远程卷，则任何其他独立索引器或索引器群集上的索引都不能使用相同的远程卷。
- 将 `maxDataSize` 保留为每个 SmartStore 索引的默认值 "auto"（750MB）。
- 每个 SmartStore 索引的 `coldPath` 设置需要一个值，即使在迁移索引的情况下，也会忽略该设置。
- 每个 SmartStore 索引的 `thawedPath` 设置需要一个值，即使该设置没有实际用途，因为您无法将数据解冻为 SmartStore 索引。请参阅“解冻数据和 SmartStore”。

部署 SmartStore

以下流程假设您正在新的索引器上部署 SmartStore。还假设您正在使用单个远程位置为索引器上的所有索引部署 SmartStore。如果您只想为某些索引部署 SmartStore，或者您为 SmartStore 索引使用多个远程位置，您可修改流程以满足您的需求。

1. 确保您满足前提条件。尤其是，请阅读以下内容：
    - SmartStore 系统要求
    - 为 SmartStore 配置 S3 远程存储，或为 SmartStore 配置 GCS 远程存储。
2. 理解 SmartStore 安全策略并在必要时在部署进程中准备实施。请参阅“采用 S3 安全策略的 SmartStore”或“采用

GCS 安全策略的 SmartStore”。
3. 安装新的 Splunk Enterprise 实例作为独立的索引器。有关如何安装 Splunk Enterprise 的信息，请阅读*《安装手册》*。
4. 新建或编辑 `$SPLUNK_HOME/etc/system/local/indexes.conf` 并指定 SmartStore 设置，如以下示例所示。

使用 S3 远程对象存储：
此示例使用 S3 远程对象存储配置 SmartStore 索引。在全局级别配置 SmartStore 相关的设置，这样所有索引器都会启用 SmartStore，它们都使用一个名为 "remote_store" 的远程存储卷。示例还会新建一个名为 "cs_index" 的索引。

```
[default]
# Configure all indexes to use the SmartStore remote volume called
# "remote_store".
# Note: If you want only some of your indexes to use SmartStore,
# place this setting under the individual stanzas for each of the
# SmartStore indexes, rather than here.
remotePath = volume:remote_store/$_index_name

# Configure the remote volume.
[volume:remote_store]
storageType = remote

# The volume's 'path' setting points to the remote storage location where
# indexes reside. Each SmartStore index resides directly below the location
# specified by the 'path' setting.
path = s3://mybucket/some/path

# The following S3 settings are required only if you're using the access and secret
# keys. They are not needed if you are using AWS IAM roles.

remote.s3.access_key = <S3 access key>
remote.s3.secret_key = <S3 secret key>
remote.s3.endpoint = https:|http://<S3 host>

# This example stanza configures a custom index, "cs_index".
[cs_index]
homePath = $SPLUNK_DB/cs_index/db
# SmartStore-enabled indexes do not use thawedPath or coldPath, but you must still specify them here.
coldPath = $SPLUNK_DB/cs_index/colddb
thawedPath = $SPLUNK_DB/cs_index/thaweddb
```

关于这些设置的详细信息，请参阅“配置 SmartStore”。另请参阅*《管理员手册》*中的 indexes.conf.spec。

使用 GCS 远程对象存储：
此示例使用 GCS 远程对象存储配置 SmartStore 索引。在全局级别配置 SmartStore 相关的设置，这样所有索引器都会启用 SmartStore，它们都使用一个名为 "remote_store" 的远程存储卷。示例还会新建一个名为 "cs_index" 的索引。

```
[default]
# Configure all indexes to use the SmartStore remote volume called
# "remote_store".
# Note: If you want only some of your indexes to use SmartStore,
# place this setting under the individual stanzas for each of the
# SmartStore indexes, rather than here.
remotePath = volume:remote_store/$_index_name

# Configure the remote volume.
[volume:remote_store]
storageType = remote

# The volume's 'path' setting points to the remote storage location where
# indexes reside. Each SmartStore index resides directly below the location
# specified by the 'path' setting.
path = gs://mybucket/some/path

# There are several ways to specify credentials. For details, see the topic,
# "SmartStore on GCS security strategies." One way to specify credentials
# is to point to a file, as shown here.
remote.gs.credential_file = credential.json
```

```
# This example stanza configures a custom index, "cs_index".
[cs_index]
homePath = $SPLUNK_DB/cs_index/db
# SmartStore-enabled indexes do not use thawedPath or coldPath, but you must still specify them here.
coldPath = $SPLUNK_DB/cs_index/colddb
thawedPath = $SPLUNK_DB/cs_index/thaweddb
```

关于这些设置的详细信息，请参阅“配置 SmartStore”。另请参阅*《管理员手册》*中的 indexes.conf.spec。

5. 重新启动索引器。
6. 测试部署。
   1. 要确认远程存储访问：
      1. 将样本文本文件放在远程存储中。
      2. 在索引器上，运行此命令以递归方式列出远程存储中显示的所有文件：

         ```
         splunk cmd splunkd rfs -- ls --starts-with volume:remote_store
         ```

      如果您运行命令时看到样本文件，则表示您具有远程存储访问权限。

   2. 验证数据是否传输到远程存储：
      1. 将一些数据发送到索引器。
      2. 等待要滚动的数据桶。如果您不想等数据桶自然滚动，您可手动滚动某些数据桶：

         ```
         splunk _internal call /data/indexes/<index_name>/roll-hot-buckets -auth <admin>:<password>
         ```
      3. 查找要上载到远程存储的温数据桶。
   3. 验证是否从远程存储传输数据：

      注意：此时，您应能够针对此数据运行正常搜索。大多数情况下，您不会传输远程存储中的任何数据，因为数据将保留在本地缓存中。因此，要验证是否从远程存储中传输数据，建议您首先将数据桶从本地缓存中逐出。
      1. 使用 POST 将数据桶从缓存中逐出至此 REST 端点：

         ```
         services/admin/cacheman/<cid>/evict
         ```

         其中 <cid> 是 bid|<bucketId>|。例如："bid|cs_index~0~7D76564B-AA17-488A-BAF2-5353EA0E9CE5|"

         注意：要获取数据桶的 bucketId，请在测试索引上运行一个搜索。例如：

         ```
         splunk search "|rest /services/admin/cacheman | search title=*cs_index*  | fields title" -auth
         <admin>:<password>
         ```
      2. 运行需要被逐出的数据桶中数据的搜索。

      索引器必须立即传输远程存储中的数据桶以运行搜索。运行搜索之后，您可检查数据桶是否重新显示在缓存中。

7. 执行任何部署独立索引器所需的所有其他配置。例如，配置索引器和转发器以及索引器和搜索头之间的连接。

后续步骤

使用 SmartStore 运行索引器之后，需要立刻关注一些配置问题。特别是：

- 编辑 $SPLUNK_HOME/etc/system/local/indexes.conf，并配置数据保留设置以确保索引器符合您需要的冻结行为。请参阅“为 SmartStore 索引配置数据保留”。

这一步极其重要，可避免不需要的数据桶冻结和可能的数据丢失。SmartStore 数据桶冻结行为和设置和非 SmartStore 行为和设置不同。

- 编辑 $SPLUNK_HOME/etc/system/local/server.conf 以对 SmartStore 相关的 server.conf 设置做出必要的更改。尤其是，配置缓存大小以满足部署需求。请参阅“配置 SmartStore 缓存管理器”。

更改后，重新启动索引器。

关于其他 SmartStore 设置的详细信息，请参阅“配置 SmartStore”。

将索引器群集上的现有数据迁移到 SmartStore

您可将索引器群集上的现有数据从本地存储迁移到远程存储。

此流程介绍如何将索引器群集上的所有索引迁移到 SmartStore。如果您只想迁移部分索引，可修改流程。索引器支持 SmartStore 和非 SmartStore 索引混合环境。

因为此进程要求群集上载大量数据，可能需要很长时间完成，并可能会对并发索引和搜索产生明显影响。

当您将索引迁移到 SmartStore 之后，就无法将索引恢复到非 SmartStore 中。

迁移数据

在两个阶段中执行迁移操作：

1. 在独立测试实例上测试 SmartStore 配置和远程连接。
2. 通过将配置应用到生产索引器群集中运行迁移。

前提条件

- 阅读：
  - SmartStore 系统要求
  - 配置 SmartStore
  - 更新通用对等节点配置和应用
  - 为 SmartStore 配置 S3 远程存储，或为 SmartStore 配置 GCS 远程存储。
  - 为每个索引选择存储位置
  - 您正在使用的远程存储服务提供商提供的文档
- 如果曾将群集从单站点迁移到多站点，您必须转换任何预先存在的单站点数据桶以遵循多站点复制和搜索策略。要执行此操作，请将管理器节点的 server.conf 文件中的 constrain_singlesite_buckets 设置更改为 "false" 并重新启动管理器节点。请参阅“配置管理器以将现有数据桶转换为多站点数据桶”。
- 群集应在较小的一侧，最多 20 个索引器。如果您想要迁移较大的群集，请联系 Splunk 专业服务。
- 小心这些配置问题：
  - 各远程卷段落的 path 设置值对于索引器群集来说必须唯一。您只能在单个群集中的各索引之间共享远程卷。换句话说，如果一个群集上的索引使用特定的远程卷，则任何其他群集或独立索引器上的索引都不能使用相同的远程卷。
  - 您必须在索引器群集中设置所有 SmartStore 索引才能使用 repfactor = auto。
  - 将 maxDataSize 保留为每个 SmartStore 索引的默认值 "auto"（750MB）。
  - 每个 SmartStore 索引的 coldPath 设置需要一个值，即使在迁移索引的情况下，也会忽略该设置。
- 每个 SmartStore 索引的 thawedPath 设置需要一个值，即使该设置没有实际用途，因为您无法将数据解冻为 SmartStore 索引。请参阅“解冻数据和 SmartStore”。
- 必要时重新配置群集以符合不支持的功能、当前限制和不兼容的设置：
  - SmartStore 不支持的功能
  - 当前 SmartStore 使用限制。关于复制因子和搜索因子相等的要求，您可在迁移后进行此更改。
  - Indexes.conf 中和 SmartStore 不兼容或有其他方面限制的设置

1. 在单独实例上测试配置

在单独实例上执行测试的目地在于：

- 测试远程存储连接。
- 验证配置。

步骤

1. 确保您符合与此测试设置相关的所有前提条件。尤其是，请阅读以下内容：
   - SmartStore 系统要求
   - 为 SmartStore 配置 S3 远程存储，或为 SmartStore 配置 GCS 远程存储。
2. 理解 SmartStore 安全策略并在必要时在部署进程中准备实施。请参阅“采用 S3 安全策略的 SmartStore”或“采用 GCS 安全策略的 SmartStore”。
3. 安装新的 Splunk Enterprise 实例。有关如何安装 Splunk Enterprise 的信息，请阅读*《安装手册》*。
4. 编辑 $SPLUNK_HOME/etc/system/local 中的 indexes.conf 以为索引指定 SmartStore 设置。这些应为一组和您之后打算用于生产部署的相同的设置。

   使用 S3 远程对象存储：
   此示例使用 S3 远程对象存储配置 SmartStore 索引。在全局级别配置 SmartStore 相关的设置，这样所有索引器都会启用 SmartStore，它们都使用一个名为 "remote_store" 的远程存储卷。示例还会新建一个名为 "cs_index" 的索引。

   ```
   [default]
   # Configure all indexes to use the SmartStore remote volume called
   # "remote_store".
   # Note: If you want only some of your indexes to use SmartStore,
   # place this setting under the individual stanzas for each of the
   # SmartStore indexes, rather than here.
   remotePath = volume:remote_store/$_index_name

   # Configure the remote volume.
   ```

```
[volume:remote_store]
storageType = remote

# The volume's 'path' setting points to the remote storage location where
# indexes reside. Each SmartStore index resides directly below the location
# specified by the 'path' setting.
path = s3://mybucket/some/path

# The following S3 settings are required only if you're using the access and secret
# keys. They are not needed if you are using AWS IAM roles.

remote.s3.access_key = <S3 access key>
remote.s3.secret_key = <S3 secret key>
remote.s3.endpoint = https:|http://<S3 host>

# This example stanza configures a custom index, "cs_index".
[cs_index]
homePath = $SPLUNK_DB/cs_index/db
# SmartStore-enabled indexes do not use thawedPath or coldPath, but you must still specify them here.
coldPath = $SPLUNK_DB/cs_index/colddb
thawedPath = $SPLUNK_DB/cs_index/thaweddb
```

关于这些设置的详细信息，请参阅“配置 SmartStore”。另请参阅*《管理员手册》*中的 indexes.conf.spec。

使用 GCS 远程对象存储：
此示例使用 GCS 远程对象存储配置 SmartStore 索引。在全局级别配置 SmartStore 相关的设置，这样所有索引器都会启用 SmartStore，它们都使用一个名为 "remote_store" 的远程存储卷。示例还会新建一个名为 "cs_index" 的索引。

```
[default]
# Configure all indexes to use the SmartStore remote volume called
# "remote_store".
# Note: If you want only some of your indexes to use SmartStore,
# place this setting under the individual stanzas for each of the
# SmartStore indexes, rather than here.
remotePath = volume:remote_store/$_index_name

# Configure the remote volume.
[volume:remote_store]
storageType = remote

# The volume's 'path' setting points to the remote storage location where
# indexes reside. Each SmartStore index resides directly below the location
# specified by the 'path' setting.
path = gs://mybucket/some/path

# There are several ways to specify credentials. For details, see the topic,
# "SmartStore on GCS security strategies." One way to specify credentials
# is to point to a file, as shown here.
remote.gs.credential_file = credential.json

# This example stanza configures a custom index, "cs_index".
[cs_index]
homePath = $SPLUNK_DB/cs_index/db
# SmartStore-enabled indexes do not use thawedPath or coldPath, but you must still specify them here.
coldPath = $SPLUNK_DB/cs_index/colddb
thawedPath = $SPLUNK_DB/cs_index/thaweddb
```

关于这些设置的详细信息，请参阅“配置 SmartStore”。另请参阅*《管理员手册》*中的 indexes.conf.spec。

5. 重新启动实例。
6. 测试部署：
    1. 要确认远程存储访问：
        1. 将样本文本文件放在远程存储中。
        2. 在 Splunk Enterprise 实例上，运行此命令以递归方式列出了远程存储中显示的所有文件。

            ```
            splunk cmd splunkd rfs -- ls --starts-with volume:remote_store
            ```

        如果您运行命令时看到样本文件，则表示您具有远程存储访问权限。

2. 验证数据是否传输到远程存储：
    1. 将一些数据发送到索引器。
    2. 等待要滚动的数据桶。如果您不想等数据桶自然滚动，您可手动滚动某些数据桶：

        ```
        splunk _internal call /data/indexes/<index_name>/roll-hot-buckets -auth <admin>:<password>
        ```
    3. 查找要上载到远程存储的温数据桶。
3. 验证是否从远程存储传输数据：

    注意：此时，您应能够针对此数据运行正常搜索。大多数情况下，您不会传输远程存储中的任何数据，因为数据将保留在本地缓存中。因此，要验证是否从远程存储中传输数据，建议您首先将数据桶从本地缓存中逐出。
    1. 使用 POST 将数据桶从缓存中逐出至此 REST 端点：

        ```
        services/admin/cacheman/<cid>/evict
        ```

        其中 `<cid>` 是 `bid|<bucketId>|`。例如："bid|cs_index~0~7D76564B-AA17-488A-BAF2-5353EA0E9CE5|"

        注意：要获取数据桶的 `bucketId`，请在测试索引上运行一个搜索。例如：

        ```
        splunk search "|rest /services/admin/cacheman | search title=*cs_index*  | fields title" -auth
        <admin>:<password>
        ```
    2. 运行需要被逐出的数据桶中数据的搜索。

    实例必须立即传输远程存储中的数据桶以运行搜索。运行搜索之后，您可检查数据桶是否重新显示在缓存中。

2. 在索引器群集上运行迁移

在此过程中，您可为 SmartStore 配置群集。此流程的目标是将所有索引上的现有温数据桶和冷数据桶迁移到 SmartStore 中。继续操作，所有新的温数据桶还将驻留在 SmartStore 中。

迁移进程完成需要一点时间。如果您有大量数据，可能会花费很长时间。预计迁移期间索引和搜索性能会下降。因此，最好是将迁移安排在索引器相对空闲的时间。

步骤

1. 确保您满足前提条件。尤其是，请阅读以下内容：
    - SmartStore 系统要求
    - 为 SmartStore 配置 S3 远程存储，或为 SmartStore 配置 GCS 远程存储。
    - SmartStore 不支持的功能
    - 当前 SmartStore 使用限制
2. 理解 SmartStore 安全策略并在必要时在部署进程中准备实施。请参阅“采用 S3 安全策略的 SmartStore”或“采用 GCS 安全策略的 SmartStore”。
3. 将所有群集节点（管理器节点、对等节点、搜索头）都升级为最新版的 Splunk Enterprise。请参阅“索引器群集升级”。
4. 确认群集处于**有效**且**完整**的状态，同时满足复制因子和搜索因子。前往“管理器节点仪表板”确认。
5. 确认没有数据桶修复任务正在进行或处于待修复状态。前往“管理器节点仪表板”，单击“索引”选项卡，然后单击“数据桶状态”按钮确认。
6. 在管理器节点上运行 `splunk enable maintenance-mode`。要确认管理器节点处于维护模式，运行 `splunk show maintenance-mode`。
7. 停止所有对等节点。当停止对等节点时，使用 `splunk stop` 命令而不是 `splunk offline`。
8. 在管理器节点上，编辑现有的 `$SPLUNK_HOME/etc/master-apps/_cluster/local/indexes.conf` 文件添加以下内容。

    请勿替换现有的 `indexes.conf` 文件。您需要保留当前设置，如索引定义设置。或者，将这些额外设置合并到现有文件中。确保移除文件中这些设置的任何其他副本。

    1. 指定 SmartStore 索引全局和卷设置。假设您已在单独实例上测试了这些设置，您可只复制单独实例中的设置，记住要添加一行 `repfactor=auto`。例如：

        使用 S3 远程对象存储：

        ```
        [default]
        # Configure all indexes to use the SmartStore remote volume called
        # "remote_store".
        # Note: If you want only some of your indexes to use SmartStore,
        # place this setting under the individual stanzas for each of the
        # SmartStore indexes, rather than here.
        remotePath = volume:remote_store/$_index_name

        repFactor = auto
        ```

```
# Configure the remote volume.
[volume:remote_store]
storageType = remote

# The volume's 'path' setting points to the remote storage location where
# indexes reside. Each SmartStore index resides directly below the location
# specified by the 'path' setting.
path = s3://mybucket/some/path

# The following S3 settings are required only if you're using the access and secret
# keys. They are not needed if you are using AWS IAM roles.

remote.s3.access_key = <S3 access key>
remote.s3.secret_key = <S3 secret key>
remote.s3.endpoint = https:|http://<S3 host>

# This example stanza configures a custom index, "cs_index".
[cs_index]
homePath = $SPLUNK_DB/cs_index/db
# SmartStore-enabled indexes do not use thawedPath or coldPath, but you must still specify them here.
coldPath = $SPLUNK_DB/cs_index/colddb
thawedPath = $SPLUNK_DB/cs_index/thaweddb
```

使用 GCS 远程对象存储：

```
[default]
# Configure all indexes to use the SmartStore remote volume called
# "remote_store".
# Note: If you want only some of your indexes to use SmartStore,
# place this setting under the individual stanzas for each of the
# SmartStore indexes, rather than here.
remotePath = volume:remote_store/$_index_name

repFactor = auto

# Configure the remote volume.
[volume:remote_store]
storageType = remote

# The volume's 'path' setting points to the remote storage location where
# indexes reside. Each SmartStore index resides directly below the location
# specified by the 'path' setting.
path = gs://mybucket/some/path

# There are several ways to specify credentials. For details, see the topic,
# "SmartStore on GCS security strategies." One way to specify credentials
# is to point to a file, as shown here.
remote.gs.credential_file = credential.json

# This example stanza configures a custom index, "cs_index".
[cs_index]
homePath = $SPLUNK_DB/cs_index/db
# SmartStore-enabled indexes do not use thawedPath or coldPath, but you must still specify them here.
coldPath = $SPLUNK_DB/cs_index/colddb
thawedPath = $SPLUNK_DB/cs_index/thaweddb
```

2. 必要时配置数据保留设置，以确保迁移后群集会遵循您希望的冻结行为。请参阅“为 SmartStore 索引配置数据保留”。

   这一步极其重要，可避免不需要的数据桶冻结和可能的数据丢失。SmartStore 数据桶冻结行为和设置和非 SmartStore 行为和设置不同。

9. 在管理器节点上，编辑 `$SPLUNK_HOME/etc/master-apps/_cluster/local/server.conf` 以对对等节点上 SmartStore 相关的 `server.conf` 设置进行必要更改。尤其是，配置缓存大小以满足部署需求。请参阅“配置 SmartStore 缓存管理器”。
10. 仅在管理器节点上，运行：

```
splunk apply cluster-bundle --answer-yes
```

11. 启动所有对等节点。稍微等待对等节点下载具有 SmartStore 设置的配置软件包。要查看配置软件包进程状态，您可运行

splunk show cluster-bundle-status 命令，详见“更新常见对等节点配置和应用”中的介绍。

12. 在管理器上运行 splunk disable maintenance-mode。要确认管理器节点不处于维护模式，运行 splunk show maintenance-mode。
13. 稍微等待对等节点开始将温数据桶和冷数据桶上载到远程存储。

    冷数据桶使用冷路径作为缓存迁移后位置。

    冷数据桶各方面功能和温数据桶相同。缓存管理器可用和管理温数据桶相同的方式管理已迁移的冷数据桶。惟一的区别在于冷数据桶将提取到冷路径位置而不是主路径位置。

14. 要确认跨索引器群集的远程存储访问，请从一个对等节点运行此命令：

    ```
    splunk cmd splunkd rfs -- ls --starts-with volume:remote_store
    ```

    此命令以递归方式列出了远程存储中显示的所有文件。应显示群集开始将温数据桶上载至远程存储。必要时，请稍等片刻，以便开始首次上载。

15. 在管理器节点上进行必要更改，以确保索引器群集复制因子和搜索因子使用的是相同值，如 3/3。
16. 要确定迁移是否完成：
    - 确认群集处于**有效**且**完整**的状态，同时满足复制因子和搜索因子。
    - 确认搜索 |rest splunk_server=idx1 ... /services/admin/cacheman |search cm:bucket.stable=0 |stats count 是否返回 0。
17. 测试 SmartStore 功能。此时，您应能够针对此数据运行正常搜索。大多数情况下，您不会传输远程存储中的任何数据，因为数据将保留在本地缓存中。要验证从远程存储中提取的数据，请执行以下操作：
    1. 在其中一个对等节点上，查找完整填充的数据桶，包含 tsidx 文件和原始数据文件。
    2. 使用此 REST 端点逐出缓存中的数据桶：

       ```
       services/admin/cacheman/<cid>/evict
       ```

       其中 <cid> 是 bid|<bucketId>|。例如："bid|cs_index~0~7D76564B-AA17-488A-BAF2-5353EA0E9CE5|"

       注意：要获取数据桶的 bucketId，请转到搜索头节点，然后在测试索引上运行一个搜索。例如：

       ```
       splunk search "|rest /services/admin/cacheman | search title=*cs_index*  | fields splunk_server, title" -auth <admin>:<password>
       ```

       结果列出了指定测试索引中的数据桶集（按 bucketId 列出）以及它们的关联对等节点。您可以基于这些信息，决定从其中一个对等节点的缓存中逐出一个数据桶。

    3. 在对等节点本地运行搜索。搜索必须需要被逐出的数据桶中的数据。

    对等节点必须立即传输远程存储中的数据桶以运行搜索。运行搜索之后，您可检查数据桶是否重新显示在缓存中。

如果您迁移期间需要重新启动群集，重新启动后，迁移会从停止位置继续。

在成功运行启用 SmartStore 群集之前，请勿重新平衡数据或删除多余的数据桶。特别是，只有在将复制因子和搜索因子设置为使用相等值并且群集已执行任何相关的数据桶修复后，才能运行这些操作。

监视迁移过程

您可以使用监视控制台监视迁移过程。请参阅“用监视控制台进行故障排除”。

您还可以从管理器节点运行端点以确定迁移状态：

```
$ splunk search "|rest /services/admin/cacheman/_metrics |fields splunk_server migration.*" -auth <admin>:<password>
```

端点会返回迁移中的数据，您可使用该数据来确定每个对等节点的进程。在此示例中，peer1 在第 8 个任务中，共计 35 个任务，因此对等节点迁移完成约 20-25%。Start_epoch 字段告诉您什么时候开始迁移，这样您可以大致推断完成时间：

```
splunk_server      migration.current_job migration.start_epoch migration.status migration.total_jobs
--------------  ---------------------- --------------------- --------------- --------------------
cluster1-manager                                               not_started
peer1.ajax.com              8                 1484942186         running               35
peer2.ajax.com              7                 1484942190         running               37
peer2.ajax.com              5                 1484942194         running               36
```

所有对等节点上的 migration.status 达到“已完成”之后，迁移结束，且 current_job 会和 total_jobs 匹配。

如果任何对等节点在迁移期间重新启动，那么迁移信息会丢失，且端点不能用于检查对等节点状态，尽管事实上迁移会恢复。甚至在迁移恢复之后，对等节点的报告状态将保留 "not_started"。

相反，您可以在重新启动的对等节点上运行以下端点：

```
"|rest /services/admin/cacheman |search cm:bucket.stable=0 |stats count"
```

计数等于剩余的上载任务数，其中上载任务表示要上载的单个数据桶，或者换句话说，来自较早端点 (total_jobs - current_jobs)。随着迁移继续，计数将减少到 0。

将独立索引器上的现有数据迁移到 SmartStore

您可将独立索引器上的现有数据从本地存储迁移到远程存储。

此流程介绍如何将索引器上的所有索引迁移到 SmartStore。如果您只想迁移部分索引，可修改流程。索引器支持 SmartStore 和非 SmartStore 索引混合环境。

因为此进程要求索引器上载大量数据，所以可能需要很长时间完成，并可能会对并发索引和搜索产生显著影响。

当您将索引迁移到 SmartStore 之后，就无法将索引恢复到非 SmartStore 中。

迁移数据

在两个阶段中执行迁移操作：

1. 在测试索引器上测试 SmartStore 配置和远程连接。
2. 通过将配置应用到生产索引器中运行迁移。

前提条件

- 阅读：
  - SmartStore 系统要求
  - 配置 SmartStore
  - 为 SmartStore 配置 S3 远程存储，或为 SmartStore 配置 GCS 远程存储。
  - 为每个索引选择存储位置
  - 您正在使用的远程存储服务提供商提供的文档
- 小心这些配置问题：
  - 各远程卷段落的 path 设置值对于索引器来说必须唯一。您只能在单个独立的索引器中的各索引之间共享远程卷。换句话说，如果一个索引器上的索引使用特定的远程卷，则任何其他独立索引器或索引器群集上的索引都不能使用相同的远程卷。
  - 将 maxDataSize 保留为每个 SmartStore 索引的默认值 "auto" (750MB)。
  - 每个 SmartStore 索引的 coldPath 设置需要一个值，即使在迁移索引的情况下，也会忽略该设置。
- 每个 SmartStore 索引的 thawedPath 设置需要一个值，即使该设置没有实际用途，因为您无法将数据解冻为 SmartStore 索引。请参阅“解冻数据和 SmartStore”。
- 必要时重新配置索引器以符合不支持的功能、当前限制和不兼容的设置列表：
  - SmartStore 不支持的功能
  - 当前 SmartStore 使用限制。
  - Indexes.conf 中和 SmartStore 不兼容或有其他方面限制的设置

1. 在测试索引器上测试配置

测试配置的目的在于：

- 测试远程存储连接。
- 验证配置。

步骤

1. 确保您符合与此测试设置相关的所有前提条件。尤其是，请阅读以下内容：
   - SmartStore 系统要求
   - 为 SmartStore 配置 S3 远程存储，或为 SmartStore 配置 GCS 远程存储。
2. 理解 SmartStore 安全策略并在必要时在部署进程中准备实施。请参阅“采用 S3 安全策略的 SmartStore”或“采用 GCS 安全策略的 SmartStore”。
3. 安装新的 Splunk Enterprise 实例。有关如何安装 Splunk Enterprise 的信息，请阅读*《安装手册》*。
4. 编辑 $SPLUNK_HOME/etc/system/local 中的 indexes.conf 以为索引指定 SmartStore 设置。这些应为一组和您之后打算用于生产部署的相同的设置。

使用 S3 远程对象存储：
此示例使用 S3 远程对象存储配置 SmartStore 索引。在全局级别配置 SmartStore 相关的设置，这样所有索引器都会启用 SmartStore，它们都使用一个名为 "remote_store" 的远程存储卷。示例还会新建一个名为 "cs_index" 的索引。

```
[default]
# Configure all indexes to use the SmartStore remote volume called
# "remote_store".
# Note: If you want only some of your indexes to use SmartStore,
# place this setting under the individual stanzas for each of the
# SmartStore indexes, rather than here.
remotePath = volume:remote_store/$_index_name

# Configure the remote volume.
[volume:remote_store]
storageType = remote

# The volume's 'path' setting points to the remote storage location where
# indexes reside. Each SmartStore index resides directly below the location
# specified by the 'path' setting.
path = s3://mybucket/some/path

# The following S3 settings are required only if you're using the access and secret
# keys. They are not needed if you are using AWS IAM roles.

remote.s3.access_key = <S3 access key>
remote.s3.secret_key = <S3 secret key>
remote.s3.endpoint = https:|http://<S3 host>

# This example stanza configures a custom index, "cs_index".
[cs_index]
homePath = $SPLUNK_DB/cs_index/db
# SmartStore-enabled indexes do not use thawedPath or coldPath, but you must still specify them here.
coldPath = $SPLUNK_DB/cs_index/colddb
thawedPath = $SPLUNK_DB/cs_index/thaweddb
```

关于这些设置的详细信息，请参阅“配置 SmartStore”。另请参阅*《管理员手册》*中的 indexes.conf.spec。

使用 GCS 远程对象存储：
此示例使用 GCS 远程对象存储配置 SmartStore 索引。在全局级别配置 SmartStore 相关的设置，这样所有索引器都会启用 SmartStore，它们都使用一个名为 "remote_store" 的远程存储卷。示例还会新建一个名为 "cs_index" 的索引。

```
[default]
# Configure all indexes to use the SmartStore remote volume called
# "remote_store".
# Note: If you want only some of your indexes to use SmartStore,
# place this setting under the individual stanzas for each of the
# SmartStore indexes, rather than here.
remotePath = volume:remote_store/$_index_name

# Configure the remote volume.
[volume:remote_store]
storageType = remote

# The volume's 'path' setting points to the remote storage location where
# indexes reside. Each SmartStore index resides directly below the location
# specified by the 'path' setting.
path = gs://mybucket/some/path

# There are several ways to specify credentials. For details, see the topic,
# "SmartStore on GCS security strategies." One way to specify credentials
# is to point to a file, as shown here.
remote.gs.credential_file = credential.json

# This example stanza configures a custom index, "cs_index".
[cs_index]
homePath = $SPLUNK_DB/cs_index/db
# SmartStore-enabled indexes do not use thawedPath or coldPath, but you must still specify them here.
coldPath = $SPLUNK_DB/cs_index/colddb
```

```
thawedPath = $SPLUNK_DB/cs_index/thaweddb
```

关于这些设置的详细信息，请参阅“配置 SmartStore”。另请参阅*《管理员手册》*中的 indexes.conf.spec。

5. 重新启动索引器。
6. 测试部署：
   1. 要确认远程存储访问：
      1. 将样本文本文件放在远程存储中。
      2. 在索引器上，运行此命令以递归方式列出远程存储中显示的所有文件：

         ```
         splunk cmd splunkd rfs -- ls --starts-with volume:remote_store
         ```

      如果您运行命令时看到样本文件，则表示您具有远程存储访问权限。

   2. 验证数据是否传输到远程存储：
      1. 将一些数据发送到索引器。
      2. 等待要滚动的数据桶。如果您不想等数据桶自然滚动，您可手动滚动某些数据桶：

         ```
         splunk _internal call /data/indexes/<index_name>/roll-hot-buckets -auth <admin>:<password>
         ```
      3. 查找要上载到远程存储的温数据桶。
   3. 验证是否从远程存储传输数据：

      注意：此时，您应能够针对此数据运行正常搜索。大多数情况下，您不会传输远程存储中的任何数据，因为数据将保留在本地缓存中。因此，要验证是否从远程存储中传输数据，建议您首先将数据桶从本地缓存中逐出。
      1. 使用 POST 将数据桶从缓存中逐出至此 REST 端点：

         ```
         services/admin/cacheman/<cid>/evict
         ```

         其中 <cid> 是 bid|<bucketId>|。例如："bid|cs_index~0~7D76564B-AA17-488A-BAF2-5353EA0E9CE5|"

         注意：要获取数据桶的 bucketId，请在测试索引上运行一个搜索。例如：

         ```
         splunk search "|rest /services/admin/cacheman | search title=*cs_index*  | fields title" -auth
         <admin>:<password>
         ```
      2. 运行需要被逐出的数据桶中数据的搜索。

      索引器必须立即传输远程存储中的数据桶以运行搜索。运行搜索之后，您可检查数据桶是否重新显示在缓存中。

2. 在生产索引器上运行迁移

在此过程中，您可为 SmartStore 配置生产索引器。此流程的目标是将所有索引上的现有温数据桶和冷数据桶迁移到 SmartStore 中。继续操作，所有新的温数据桶还将驻留在 SmartStore 中。

迁移进程完成需要一点时间。如果您有大量数据，可能会花费很长时间。预计迁移期间索引和搜索性能会下降。因此，最好是将迁移安排在索引器相对空闲的时间。

步骤

1. 确保您满足前提条件。尤其是，请阅读以下内容：
   - SmartStore 系统要求
   - 为 SmartStore 配置 S3 远程存储，或为 SmartStore 配置 GCS 远程存储。
   - SmartStore 不支持的功能
   - 当前 SmartStore 使用限制
2. 理解 SmartStore 安全策略并在必要时在部署进程中准备实施。请参阅“采用 S3 安全策略的 SmartStore”或“采用 GCS 安全策略的 SmartStore”。
3. 将索引器升级到 Splunk Enterprise 的最新版本。
4. 停止索引器。
5. 编辑现有的 $SPLUNK_HOME/etc/system/local/indexes.conf 文件添加以下内容。

   请勿替换现有的 indexes.conf 文件。您需要保留当前设置，如索引定义设置。或者，将这些额外设置合并到现有文件中。确保移除文件中这些设置的任何其他副本。

   1. 指定 SmartStore 索引全局和卷设置。假设您已在测试实例上测试了这些设置，您可只复制测试实例中的设置。例如：

      使用 S3 远程对象存储：

      ```
      [default]
      # Configure all indexes to use the SmartStore remote volume called
      ```

```
# "remote_store".
# Note: If you want only some of your indexes to use SmartStore,
# place this setting under the individual stanzas for each of the
# SmartStore indexes, rather than here.
remotePath = volume:remote_store/$_index_name

# Configure the remote volume.
[volume:remote_store]
storageType = remote

# The volume's 'path' setting points to the remote storage location where
# indexes reside. Each SmartStore index resides directly below the location
# specified by the 'path' setting.
path = s3://mybucket/some/path

# The following S3 settings are required only if you're using the access and secret
# keys. They are not needed if you are using AWS IAM roles.

remote.s3.access_key = <S3 access key>
remote.s3.secret_key = <S3 secret key>
remote.s3.endpoint = https:|http://<S3 host>

# This example stanza configures a custom index, "cs_index".
[cs_index]
homePath = $SPLUNK_DB/cs_index/db
# SmartStore-enabled indexes do not use thawedPath or coldPath, but you must still specify them here.
coldPath = $SPLUNK_DB/cs_index/colddb
thawedPath = $SPLUNK_DB/cs_index/thaweddb
```

使用 GCS 远程对象存储：

```
[default]
# Configure all indexes to use the SmartStore remote volume called
# "remote_store".
# Note: If you want only some of your indexes to use SmartStore,
# place this setting under the individual stanzas for each of the
# SmartStore indexes, rather than here.
remotePath = volume:remote_store/$_index_name

# Configure the remote volume.
[volume:remote_store]
storageType = remote

# The volume's 'path' setting points to the remote storage location where
# indexes reside. Each SmartStore index resides directly below the location
# specified by the 'path' setting.
path = gs://mybucket/some/path

# There are several ways to specify credentials. For details, see the topic,
# "SmartStore on GCS security strategies." One way to specify credentials
# is to point to a file, as shown here.
remote.gs.credential_file = credential.json

# This example stanza configures a custom index, "cs_index".
[cs_index]
homePath = $SPLUNK_DB/cs_index/db
# SmartStore-enabled indexes do not use thawedPath or coldPath, but you must still specify them here.
coldPath = $SPLUNK_DB/cs_index/colddb
thawedPath = $SPLUNK_DB/cs_index/thaweddb
```

2. 必要时配置数据保留设置，以确保迁移后索引器会遵循您希望的冻结行为。请参阅“为 SmartStore 索引配置数据保留”。

   这一步极其重要，可避免不需要的数据桶冻结和可能的数据丢失。SmartStore 数据桶冻结行为和设置和非 SmartStore 行为和设置不同。

6. 编辑 `$SPLUNK_HOME/etc/system/local/server.conf` 以对 SmartStore 相关的 `server.conf` 设置做出必要的更改。尤其是，配置缓存大小以满足部署需求。请参阅“配置 SmartStore 缓存管理器”。
7. 启动索引器。

8. 稍微等待索引器开始将温数据桶和冷数据桶上载到远程存储。

   冷数据桶使用冷路径作为缓存迁移后位置。

   冷数据桶各方面功能和温数据桶相同。缓存管理器可用和管理温数据桶相同的方式管理已迁移的冷数据桶。惟一的区别在于冷数据桶将提取到冷路径位置而不是主路径位置。

9. 要确认远程存储访问，运行此命令：

   ```
   splunk cmd splunkd rfs -- ls --starts-with volume:remote_store
   ```

   此命令以递归方式列出了远程存储中显示的所有文件。应显示索引器开始将温数据桶上载至远程存储。必要时，请稍等片刻，以便开始首次上载。

10. 要确认迁移是否完成，请参阅“监视迁移进程”。
11. 测试 SmartStore 功能。此时，您应能够针对此数据运行正常搜索。大多数情况下，您不会传输远程存储中的任何数据，因为数据将保留在本地缓存中。要验证从远程存储中提取的数据，请执行以下操作：
    1. 在索引器上，查找完整填充的数据桶，包含 tsidx 文件和原始数据文件。
    2. 使用 POST 将数据桶从缓存中逐出至此 REST 端点：

       ```
       services/admin/cacheman/<cid>/evict
       ```

       其中 <cid> 是 bid|<bucketId>|。例如："bid|cs_index~0~7D76564B-AA17-488A-BAF2-5353EA0E9CE5|"

       注意：要获取数据桶的 bucketId，请在测试索引上运行一个搜索。例如：

       ```
       splunk search "|rest /services/admin/cacheman | search title=*cs_index*  | fields title" -auth <admin>:<password>
       ```
    3. 在索引器上本地运行搜索。搜索必须需要被逐出的数据桶中的数据。

    索引器必须立即传输远程存储中的数据桶以运行搜索。运行搜索之后，您可检查数据桶是否重新显示在缓存中。

如果您迁移期间需要重新启动索引器，重新启动后，迁移会从停止位置继续。

监视迁移过程

您可以使用监视控制台监视迁移过程。请参阅“用监视控制台进行故障排除”。

您还可以从索引器运行端点以确定迁移状态：

```
$ splunk search "|rest /services/admin/cacheman/_metrics |fields splunk_server migration.*" -auth <admin>:<password>
```

端点会返回迁移中的数据，您可使用该数据来确定索引器的进程。

如果索引器在迁移期间重新启动，那么迁移信息会丢失，且端点不能用于检查索引器状态，尽管事实上迁移会恢复。甚至在迁移恢复之后，索引器的报告状态仍旧为 "not_started"。

相反，您可以在索引器上运行以下端点：

```
"|rest /services/admin/cacheman |search cm:bucket.stable=0 |stats count"
```

计数等于剩余的上载任务数，其中上载任务表示要上载的单个数据桶。随着迁移继续，计数将减少到 0。

启用 SmartStore 索引

启动是一个将 SmartStore 索引传输到新的索引器群集或独立索引器的过程。首先，关闭有 SmartStore 索引的旧的索引器群集或索引器。然后将新的索引器群集或独立的索引器指向取消配置的群集或索引器上的 SmartStore 索引之前使用的远程存储位置。新的群集或索引器会继承旧群集或索引器的 SmartStore 数据桶。

例如，如果您想要替换索引器群集，您可关闭旧的群集然后启动新的群集。新的群集会继承旧群集的所有数据桶。如果您需要升级运行索引器群集的计算机，这样会有用。

同样，您可以通过启动用于可伸缩性和高可用性目的，以将 SmartStore 索引从独立索引器移动到索引器群集。

多个索引器群集或独立的索引器无法访问相同的远程卷。换句话说，indexes.conf 中任何 path 设置的值对单独的运行索引器或群集来说必须是唯一。请勿在多个索引器或群集中共享 path 设置。

因此，您必须关闭当前从特定远程卷访问数据桶的群集或独立索引器，才能启动新的群集或索引器访问相同远程卷的数据桶。

启动类型

您可以将 SmartStore 索引从一个索引器群集或独立索引器启动到另一个索引器群集或独立索引器。换句话说，支持以下启动类型：

- 索引器群集到索引器群集
- 独立索引器到独立索引器
- 独立索引器到索引器群集
- 索引器群集到独立索引器

在索引器群集上启动索引

您可以在索引器群集上启动旧群集或独立索引器中的索引。

首次启动时如何启动索引

您可以启动所有索引或只启动索引子集。请参阅“在新的索引器群集上部署 SmartStore”获取有关如何使用 indexes.conf 全局或在单个索引上配置 SmartStore 的信息。

您可以在群集首次启动时或在其生命周期的后期启动 SmartStore 索引。这部分介绍如何在首次启动时启动索引。

在新的索引器群集上启动 SmartStore 索引：

1. 在确保所有本地数据（包括任何热数据桶中的数据）均可在远程存储上使用后关闭旧的群集或独立索引器。
2. 启动新的群集并为 SmartStore 配置索引。按照索引器群集中部署 SmartStore 的流程。
   请注意以下事项：
   - 配置 indexes.conf 时，为您想要启动的所有 SmartStore 索引添加段落。
   - 确认您已全局设置 repFactor=auto，这样适用于群集中的所有索引，包括您打算启动的索引。
   - 配置 path 属性以指向和旧群集或独立索引器一样的远程存储位置。
   - 维护旧群集或独立索引器使用的远程存储加密和访问设置。
   - 您启动的所有索引必须在新群集上启用 SmartStore。您无法使用启动将索引还原为非 SmartStore。

启动时

管理器节点会在以下任意条件时启用启动：

- 在索引器群集启动后，已向管理器节点注册对等节点的复制因子数。
- 管理器节点通过配置软件包推送方法将新的启动配置推送到对等节点后。这允许您在群集已启动并运行后启动 SmartStore 索引，或者在群集生命周期的不同点有选择地启动 SmartStore 索引。

启动流程

启动期间，管理器节点会协调发现过程：

1. 管理器节点会获取对等节点中所有 SmartStore 索引的列表。
2. 在每个索引上继续启动：
   1. 管理器节点会分配一个对等节点以获取远程存储中索引的所有数据桶列表。
   2. 对等节点会批量将数据桶列表返回到管理器节点中。

      通过管理器节点的 server.conf 文件中的 recreate_index_fetch_bucket_batch_size 属性控制批量大小。

   3. 管理器节点会使用列表跨所有可用对等节点随机地均匀分配数据桶。
   4. 管理器节点会为每个对等节点提供一组目标对等节点，数量和复制因子相等。
   5. 每个对等节点会从远程存储中下载分配的数据桶元数据。
   6. 对等节点会使用下载的元数据更新索引的 .bucketManifest 文件并为其分配的数据桶创建空白目录。每个数据桶的主要标记会驻留在从远程存储中下载数据桶元数据的对等节点中。
   7. 对等节点会将其主要数据桶的元数据复制到目标对等节点中以满足复制因子。

启动过程会很快结束，因为只下载和复制数据桶元数据。启动过程中不会下载数据桶内容，但是稍后如有需要会进行下载以便于搜索。

您可以使用监视控制台监视启动过程。请参阅“用监视控制台进行故障排除”。

在独立索引器上启动索引

您可以在独立索引器上启动旧独立索引器或群集中的索引。

您不能对同一个远程存储运行多个独立索引器。只有单个独立索引器才能访问特定的远程存储。

首次启动时如何启动索引

您可启动远程存储中的所有索引或只启动一小组索引。请参阅“在新的独立索引器上部署 SmartStore”获取有关如何使用 indexes.conf 全局或在单个索引上配置 SmartStore 的信息。

您可以在索引器首次启动时或在其生命周期的后期启动 SmartStore 索引。这部分介绍如何在首次启动时启动索引。稍后启动群集的过程是向现有索引器添加索引的简单变体。

要在新的索引器上启动 SmartStore 索引：

1. 在确保所有本地数据（包括任何热数据桶中的数据）均可在远程存储上使用后关闭旧的独立索引器或群集。
2. 启动新的索引器并为 SmartStore 配置索引。遵循“在新的独立索引器上部署 SmartStore”中的流程。
   请注意以下事项：
   - 配置 indexes.conf 时，为您想要启动的所有 SmartStore 索引添加段落。
   - 配置 path 属性以指向和旧群集或独立索引器一样的远程存储位置。
   - 维护旧群集或独立索引器使用的远程存储加密和访问设置。
   - 您启动的所有索引必须在新索引器上启用 SmartStore。您无法使用启动将索引还原为非 SmartStore。
3. 重新启动索引器。

索引器重启后开始启动。

启动流程

启动期间，索引器会执行发现进程。在每个索引上继续启动：

1. 索引器可获取远程存储中索引的所有数据桶列表。
2. 索引器会从远程存储中下载索引的数据桶元数据。
3. 索引器会使用下载的元数据更新索引的 .bucketManifest 文件并为数据桶新建空白目录。

启动过程会很快结束，因为只下载数据桶元数据。启动过程中不会下载数据桶内容，但是稍后如有需要会进行下载以便于搜索。

您可以使用监视控制台监视启动过程。请参阅“用监视控制台进行故障排除”。

管理 SmartStore

配置 SmartStore

SmartStore 配置设置驻留在以下三个文件中：

- indexes.conf
- server.conf
- limits.conf

Indexes.conf 中的 SmartStore 设置

indexes.conf 中的 SmartStore 设置启用和控制 SmartStore 索引。

您可为索引器的所有索引启用 SmartStore，或者你可基于每个索引启用 SmartStore，允许在同一索引器上混合使用 SmartStore 和非 SmartStore 索引。

当您在索引器群集对等节点上配置这些设置时，您必须通过配置软件包方法部署设置。和 indexes.conf 中的所有设置一样，所有对等节点的 SmartStore 设置必须相同。

此表列出了 indexes.conf SmartStore 相关的主要设置：

SmartStore indexes.conf 设置	段落级别	描述
remotePath = <root path for remote volume>	索引或全局	启用 SmartStore 并为索引卷设置远程路径。请参阅“在新的索引器群集上部署 SmartStore”。
storageType = remote	卷	将卷的存储类型设为远程。请参阅“在新的索引器群集上部署 SmartStore”。
path = <scheme>://<remote-location-specifier>	卷	设置索引驻留的远程存储位置。 每个 SmartStore 索引都直接驻留在路径设置指定的位置下。<scheme> 识别受支持的远程存储系统类型，如 S3 或 GS (GCS)。<remote-location-specifier> 是特定于远程存储系统的字符串，用于指定索引在远程系统内的位置。请参阅“在新的索引器群集上部署 SmartStore”。 **警告：**各远程卷段落的 path 设置值对索引器群集或独立索引器必须唯一。您只能在单个群集或独立索引器中的各索引之间共享远程卷。例如，如果一个群集上的索引使用特定的远程卷，则任何其他群集或独立索引器上的索引都不能使用相同的远程卷。
maxGlobalDataSizeMB = <integer>	索引或全局	确定数据桶冻结行为。设置 SmartStore 索引的温数据桶和冷数据桶可以占用的最大空间量。超出最大空间量时，最旧的数据桶会冻结。请参阅“为 SmartStore 索引配置数据保留”。
maxGlobalRawDataSizeMB= <integer>	索引或全局	确定数据桶冻结行为。设置 SmartStore 索引的温数据桶和冷数据桶可以占用的最大累计原始数据量。超出最大空间量时，最旧的数据桶会冻结。请参阅“为 SmartStore 索引配置数据保留”。
hotlist_recency_secs = <integer>	index	根据数据桶的时长指定时间周期，该期限为缓存管理器尝试保护最近的数据桶不被逐出的期限。此设置会在每个索引级别操作，然而 server.conf 中的设置版本可在所有索引中操作。请参阅“配置 SmartStore 缓存管理器”。
hotlist_bloom_filter_recency_hours = <integer>	index	根据数据桶的时长指定时间期限，该期限为缓存管理器尝试保护数据桶的非日志和非 tsidx 文件（如 bloomfilter 文件）不被逐出的期限。此设置会在每个索引级别操作，然而 server.conf 中的设置版本可在所有索引中操作。请参阅“配置 SmartStore 缓存管理器”。

在 indexes.conf 中，您还可针对远程存储类型配置不同的设置。例如，有很多以 remote.s3 开头的设置，如 remote.s3.access_key 和 remote.s3.secret_key。这些设置特定于 S3。同样，有很多以 remote.gs 开头的设置。这些设置特定于 GCS。您可在配置远程卷的段落中配置这些设置。有关这些设置和其他 indexes.conf 设置的详细信息，请参阅 indexes.conf.spec。

指定以 remote.s3 开头的设置时，您必须在 "s3" 中使用小写 "s"。大写 "S"，例如 remote.S3 不是有效的替换，会导致对设置值的任何尝试更新都不生效。

Indexes.conf 中非 SmartStore 特定的设置

您必须为所有 SmartStore 索引指定这些配置：

- repfactor = auto，针对索引器群集对等节点（非独立索引器）上的索引
- maxDataSize = auto。这是默认值（750MB），因此通常不需要显式设置。
- homePath 需要路径值。这是本地存储的位置，热数据桶和缓存的温数据桶驻留在这个位置。
- coldPath 需要路径值，即使这些值在迁移的索引中被忽略。
- thawedPath 需要路径值，即使该设置没有实际用途，因为您无法将数据解冻为 SmartStore 索引。请参阅“解冻数据和 SmartStore”。

Indexes.conf 中和 SmartStore 不兼容或有其他方面限制的设置

SmartStore 索引器和索引器群集支持在非 SmartStore 索引器和索引器群集中可用的功能子集。与不可兼容的功能相关的设置有一些特别的注意事项。

此外，和非 SmartStore 索引器和索引器群集不同，SmartStore 索引器和索引器群集可处理某些需求，如数据保留。相关设置也不一样。

有关使用 SmartStore 时不支持或限制的一些功能的信息，请参阅：

- SmartStore 不支持的功能
- 当前 SmartStore 使用限制

必须取消设置以下 indexes.conf 设置：

- bloomHomePath
- summaryHomePath
- tstatsHomePath

以下 indexes.conf 设置必须保留默认值：

- createBloomfilter。不要更改默认值 true。
- enableOnlineBucketRepair。不要更改默认值 true。
- isReadOnly。不要更改默认值 false。
- enableTsidxReduction。不要更改默认值 false。
- maxDataSize。不要更改默认值 auto（建议）。

SmartStore 会忽略以下 indexes.conf 设置：

- maxWarmDBCount
- maxTotalDataSizeMB
- warmToColdScript
- homePath.maxDataSizeMB
- coldPath.maxDataSizeMB
- maxVolumeDataSizeMB

journalCompression 设置可以设置为 gzip（默认设置）或 zstd。

路径设置和 SmartStore

SmartStore 使用 remotePath 和 path 设置识别远程存储上索引存储的位置。

所有其他路径设置标识本地缓存上的位置，而不是远程存储。这些设置包括：

- homePath
- coldPath
- thawedPath
- bloomHomePath
- summaryHomePath
- tstatsHomePath

有关每个设置的配置和使用说明，请参阅本主题的其他部分。

server.conf 中的 SmartStore 设置

server.conf 中 SmartStore 相关的设置可控制索引器的行为，包括缓存管理器的功能。此表列出了配置最频繁的设置。

SmartStore server.conf 设置	段落级别	描述

eviction_policy = <string>	[cachemanager]	设置用于确定缓存管理器下一个逐出的数据桶的策略。默认为 "lru"。请参阅“配置 SmartStore 缓存管理器”。
max_cache_size = <integer>	[cachemanager]	指定缓存可在磁盘分区上占用的最大空间（MB）。请参阅“配置 SmartStore 缓存管理器”。
eviction_padding = <integer>	[cachemanager]	指定缓存用作开始逐出数据、超出 minFreeSpace 的空间大小（MB）。请参阅“配置 SmartStore 缓存管理器”。
hotlist_recency_secs = <integer>	[cachemanager]	根据数据桶的时长指定时间周期，该期限为缓存管理器尝试保护最近的数据桶不被逐出的期限。此设置可跨索引器在全局级别操作，然而 indexes.conf 中的设置版本可在每个索引级别中操作。请参阅“配置 SmartStore 缓存管理器”。
hotlist_bloom_filter_recency_hours = <integer>	[cachemanager]	根据数据桶的时长指定时间期限，该期限为缓存管理器尝试保护数据桶的非日志和非 tsidx 文件（如 bloomfilter 文件）不被逐出的期限。此设置可跨索引器在全局级别操作，然而 indexes.conf 中的设置版本可在每个索引级别中操作。请参阅“配置 SmartStore 缓存管理器”。
各种设置	[clustering]	包括用于控制索引器群集中远程数据桶操作的各种设置（主要针对低级别）。
cleanRemoteStorageByDefault = <bool>	[general]	使用 splunk clean eventdata 命令清除远程索引。默认为 "false"。

有关这些设置和其他 server.conf 设置的详细信息，请参阅 server.conf.spec。

在对等节点上配置大部分 SmartStore server.conf 设置（包括所有常用设置）。在管理器节点上配置少数很少更改的设置。在对等节点上配置的所有设置在所有对等节点中必须相同。

limits.conf 中的 SmartStore 设置

limits.conf 文件包含几个与搜索相关的低级设置，例如 bucket_localize_max_timeout_sec。这些设置主要与本地化数据桶过程有关。在不了解基本过程的情况下，不要更改设置。

关于 limits.conf 设置的详细信息，请参阅 limits.conf.spec。

配置 SmartStore 缓存管理器

缓存管理器通过智能管理本地缓存来最大化搜索效率。有利于在缓存中保留很有可能参与未来搜索的数据桶和文件的副本。当缓存填满时，缓存管理器会删除或“逐出”最不可能参与未来搜索的数据桶副本。

由于缓存管理器仅删除数据桶的缓存副本，因此逐出过程不会导致数据丢失。管理器副本仍会保留在远程存储中。

有关缓存管理器如何操作的详细信息，请参阅“SmartStore 缓存管理器”。

缓存管理器设置驻留在 server.conf 中的 [cachemanager] 段落中。如果是索引器群集，在各对等节点上配置缓存管理器。

缓存管理器在全局级别跨索引器上的所有索引操作。除了新近设置外，您无法在每个索引器上配置缓存管理器。

设置缓存逐出策略

server.conf 中的 eviction_policy 设置决定缓存逐出策略。

逐出策略	描述
lru（默认）	逐出最近很少使用的数据桶。
lruk	删除最近使用最少的数据桶，追踪最后 K 对常用数据桶的参考，其中 K=3。
clock	除非最近访问过，否则首先逐出具有最早事件的数据桶。
lrlt	首先逐出具有最早事件的数据桶。
random	随机逐出数据桶。
noevict	不逐出。

未咨询 Splunk 支持之前，请不要更改 eviction_policy 的默认值 "lru"。

根据缓存磁盘分区的占用情况启动逐出

server.conf 中的这些设置会根据缓存磁盘分区的占用情况启动逐出：

- max_cache_size 设置指定包含缓存的磁盘分区的最大占用空间（以 MB 为单位）。
- minFreeSpace 设置指定分区的最小可用空间（以 MB 为单位）。
- eviction_padding 设置控制缓存管理器保护的超出 minFreeSpace 值的额外空间量（以 MB 为单位）。

minFreeSpace 设置严格来说并不是特定于缓存的设置，因此不会驻留在 [cachemanager] 段落中，但是该设置仍有助于确定缓存大小限制。

当缓存分区上的占用空间超过 max_cache_size 时，或者分区的可用空间低于（minFreeSpace + eviction_padding）时，缓存管理器开始逐出数据。

根据数据新近程度设置缓存保留期

您可防止最近索引的数据被逐出。您可以通过以下两种方式使用此功能：

- 在全局级别（跨所有索引），支持最近索引的数据而非最近使用的数据。
- 在每个索引上，支持重要索引中的数据而不是不重要索引中的数据。

要根据数据新近程度设置缓存保存时长，请使用 hotlist_recency_secs 和 hotlist_bloom_filter_recency_hours 设置。这些设置用于覆盖逐出策略。您可以在全局或按索引级别设置这些设置的范围。

hotlist_recency_secs 设置

hotlist_recency_secs 设置会使缓存管理器保护包含最近数据的数据桶而不是其他数据桶。该设置根据年龄确定的温数据桶的缓存保留期。当需要逐出时，除非所有其他数据桶已被逐出，否则缓存管理器将不会逐出数据桶，直到数据桶达到配置的时间。

缓存管理器会尝试延后将数据桶逐出，直到数据桶中所有数据的存在时间都大于所设置的值。此设置默认为 86400 秒，或 24 小时。

数据桶时长（或“新近程度”）的确定方式为：用当前时间减去数据桶的最近事件数据的时间。例如，如果当前时间（以 UTC epoch 时间表示）是 **1567891234**（CEST 2019 年 9 月 7 日 23:20:34），并且数据桶名为 db_**1567809123**_**1557891234**_10_8A21BEE9-60D4-436B-AA6D-21B68F631A8B（在 CEST 2019 年 5 月 15 日 05:33:54 和 CEST 2019 年 9 月 7 日 00:32:03 之间），即表示数据桶中最近事件的时间为 **1567809123**（CEST 2019 年 9 月 7 日 00:32:03），则数据桶的时长为 82111 秒（约 23 小时）。

确保缓存的大小足以处理此设置的值。否则，缓存逐出将无法发挥最佳作用。换句话说，请勿仅基于此设置将其配置为某个大小，而此大小会导致缓存保留接近或超过缓存大小的数据桶量。另外，请考虑数据引入率和搜索的典型时间跨度，以确定最近的数据桶应在缓存中保留多长时间。

最佳做法是，先为此设置设一个较低值，然后随时间的推移进行调整。例如，如果缓存大小为 100 GB，而且您通常在 24 小时内会将 10 GB 的新数据桶添加到索引器中，则将此设置配置为 172800 秒（48 小时）意味着缓存管理器会尝试随时在缓存中保留 20 GB 的最新数据桶。

hotlist_bloom_filter_recency_hours 设置

hotlist_bloom_filter_recency_hours 设置会保护某些小的元数据文件不被逐出，如 bloomfilter 文件。缓存管理器有时可通过检查这些元数据文件，在处理搜索请求时，无需从远程存储中提取较大的数据桶文件，如原始数据日志和 tsidx 文件。请参阅“SmartStore 缓存管理器”。

hotlist_bloom_filter_recency_hours 设置会影响小的温数据桶文件的缓存保留期。缓存管理器会试图推迟逐出非日志数据桶文件和非 tsidx 数据桶文件（如 bloomfilter 文件），直到数据桶最晚时间和当前时间之间的间隔超出此设置。此设置默认为 360 小时，或 15 天。

bloomfilter 文件的新近程度取决于其数据桶的新近程度，并且其计算方式与针对 hotlist_recency_secs 所述的方式相同。

此设置与 hotlist_recency_secs 配合使用，后者用于被配置为较短的时长。如果 hotlist_recency_secs 导致数据桶被逐出，则数据桶的 bloomfilter 和关联文件会继续保留在缓存中，直到达到 hotlist_bloom_filter_recency_hours 配置的时长。因此，数据桶会保留在高速缓存中，但没有日志和 tsidx 文件。

全局或为单个索引配置新近程度

当您在全局范围内配置这些设置时，这些设置会覆盖逐出策略，该策略默认支持最近搜索的数据桶。例如，如果在全局范围内将 hotlist_recency_secs 设为 604800（7 天），缓存管理器会尝试保留包含七天内的数据的数据桶。管理器会逐出旧的数据桶，即使这些旧的数据桶最近搜索频率更高。如果没有更旧的数据桶要逐出，缓存管理器将仅逐出包含七天内数据的数据桶。

您可通过在每个索引上配置新近设置，支持重要索引中的数据而不是不重要索引中的数据。由于所有 SmartStore 索引共享缓

存并遵循全局逐出策略，因此每个索引新近程度设置提供了保留重要索引中数据的时长长于不重要索引中数据的唯一方式。

例如，如果您有一个包含重要数据的索引（如 ES threat_activity 索引）和一个所含数据没那么重要的索引（如默认的 _internal 索引），您可将 threat_activity 的 hotlist_recency_secs 设为 5184000（60 天），同时保留 _internal 的默认设置 86400（1 天）。这样操作，您可使缓存管理器保留 threat_activity 数据桶而不是 _internal 数据桶，同时降低缓存从远程存储中提取数据以处理 threat_activity 搜索的可能性。

为所有索引全局配置新近程度

要为所有 SmartStore 索引全局配置 hotlist_recency_secs 和 hotlist_bloom_filter_recency_hours 设置，您必须在 server.conf 中的 [cachemanager] 段落中进行设置。

您可按索引覆盖全局设置。

为单个索引配置新近程度

要按索引配置 hotlist_recency_secs 和 hotlist_bloom_filter_recency_hours 设置，您必须在 indexes.conf 的每个索引段落中进行设置。

如果您没有针对特定的 SmartStore 索引配置设置，该索引会继承 server.conf 的全局值。

设置最大下载速度和上载速度

server.conf 中的 max_concurrent_downloads 设置可指定可同时从远程存储中下载的最大数据桶数量。默认值为 8。

server.conf 中的 max_concurrent_uploads 设置可指定可同时上载至远程存储的最大数据桶数量。默认值为 8。

为 SmartStore 索引配置数据保留

使用和非 SmartStore 索引相似的设置配置 SmartStore 索引的数据保留策略。

在索引器群集上，SmartStore 索引的数据保留策略是在群集范围内进行管理。索引中的数据桶满足冻结条件之后，群集会从远程存储和存在副本的任何本地缓存中将数据桶完全从系统中删除。

因为 SmartStore 索引不支持冷数据桶状态，已迁移数据桶除外，数据桶直接从温数据桶滚动到冻结数据桶。

有关数据桶冻结和归档的一般信息，请参阅“设置退休和归档策略”和“将索引的数据归档”。本主题中的大部分材料都与所有索引、SmartStore 或非 SmartStore 相关。本主题介绍了其中的区别。

数据保留策略

使用这些 indexes.conf 设置为 SmartStore 索引配置数据保留策略：

- maxGlobalDataSizeMB
- maxGlobalRawDataSizeMB
- frozenTimePeriodInSecs

只要达到这些限制中的任意一项，就会发生数据桶冻结。如果未能正确配置这些设置，可能会导致数据意外丢失。例如，如果在 frozenTimePeriodInSecs 之前达到了 maxGlobalDataSizeMB，则数据桶将在配置的时间周期过去之前滚动到冻结状态。如果您需要将数据保留在系统中一段特定的时间，确保其他设置不会抢占您的 frozenTimePeriodInSecs 设置。

配置数据保留限制时，确保符合任何重要的保留条件。

如果是索引器群集，数据保留设置（如所有的 indexes.conf 设置）必须和所有对等节点一样。使用配置软件包方法将管理器节点中的设置分布到对等节点中，详见“更新常用对等节点配置和应用”中的介绍。

这些仅可用于非 SmartStore 索引的设置对 SmartStore 索引没有影响：

- maxTotalDataSizeMB
- maxWarmDBCount

maxGlobalDataSizeMB

maxGlobalDataSizeMB 设置可为 SmartStore 索引中的所有温数据桶和冷数据桶指定最大大小 (MB)。当一组索引的温数据桶和冷数据桶大小超出此值时，系统会将最旧的数据桶冻结，直到数据桶大小再次小于此值。

索引的温数据桶和冷数据桶总大小约等于索引在远程存储中占用的大小。计算大小时请注意以下方面：

- 在每个索引上均适用。
- 如果是索引器群集，在群集中的所有对等节点中适用。
- 如果是独立的索引器，仅适用于该索引器。根据性质，独立索引器管理各自的数据保留。
- 包括驻留在远程存储上的所有数据桶的大小总和，连同最近从热状态滚动到温状态的数据桶和等待上载至远程存储的任何数据桶。
- 只包含每个数据桶的一个副本。如果索引器上存在数据桶的两个副本，则计算大小时不包含在内。例如，如果数据桶同时存在于远程存储和索引器的本地缓存上，计算时会忽略本地缓存上的副本。
- 只计算数据桶本身大小。不会计算任何相关文件的大小，如报表加速和数据模型加速摘要。

如果索引占用的温数据桶和冷数据桶的总大小超出 maxGlobalDataSizeMB，则会冻结索引中最旧的数据桶。例如，假设将索引的 maxGlobalDataSizeMB 设为 5000，索引的温数据桶和冷数据桶占用 4800MB。如有有 750MB 的热数据桶滚动为温数据桶，那么索引大小就会超出 maxGlobalDataSizeMB，导致数据桶冻结。群集会冻结索引上最旧的数据桶，直到温数据桶和冷数据桶的总大小低于 maxGlobalDataSizeMB。

您可在索引适用的段落下设置此值。要为所有索引指定相同值，您可在全局段落级别设置值。但是，在这种情况下，值仍单独适用于各索引。即，如果您在全局段落级别将 maxGlobalDataSizeMB 设为 5000MB，那么 indexA 的最大值为 5000MB，indexB 最大值为 5000MB，依此类推。

此设置默认为 0，表示不限制索引上温数据桶和冷数据桶的可占用的空间量。

maxGlobalRawDataSizeMB

maxGlobalRawDataSizeMB 设置可为 SmartStore 索引中的所有温数据桶中驻留的原始数据的指定最大大小 (MB)。当一组索引的温数据桶中索引的原始数据大小超出此值时，系统会将最旧的数据桶冻结，直到数据桶大小再次小于此值。

原始数据大小是索引器引入该数据时测量的未压缩的数据大小。并不是驻留在数据桶中压缩的原始数据日志文件的大小。

如果要根据最初引入的原始数据量保留数据，而不是基于数据的使用期限或数据占用的存储大小，此设置非常有用。例如，您可能需要索引保留最新的 10TB 引入的原始数据。由于索引过程压缩引入的数据并建立索引，因此存储在磁盘上的索引数据的大小可能与其原始大小明显不同。

如果是索引器群集，maxGlobalRawDataSizeMB 计算为跨所有对等节点的索引引入以及当前驻留在温数据桶或冷数据桶中的原始数据总量。只计算引入的数据计数，因此，不会通过数据复制来增加原始数据量。例如，在具有复制因子 3 的三个对等节点群集中，如果 peer1 引入 300MB 原始数据，peer2 引入 400MB 原始数据，peer3 引入 500MB 原始数据，那么驻留在群集中的原始数据总量为 1200MB，而不是 3600MB。

计算大小时注意这些关键方面：

- 使用的是原始数据大小，即索引器引入数据时未压缩的数据大小。
- 在每个索引上均适用。
- 仅适用于温数据桶和冷数据桶。热数据桶数据遭忽略
- 如果是索引器群集，在群集中的所有对等节点中适用。

如果驻留在索引的温数据桶的原始引入数据总大小超出 maxGlobalRawDataSizeMB，则会冻结索引中最旧的数据桶。例如，假设将索引的 maxGlobalRawDataSizeMB 设为 5000 (MB)，索引的温数据桶占用 4800MB 的原始数据。如果占用 500MB 原始数据的热数据桶滚动到温数据桶，那么索引中的原始数据量会超出 maxGlobalRawDataSizeMB，触发数据桶冻结。系统会冻结索引上最旧的数据桶，直到温数据桶中驻留的原始数据总量低于 maxGlobalRawDataSizeMB。

您可在索引适用的段落下设置此值。要为所有索引指定相同值，您可在全局段落级别设置值。但是，在这种情况下，值仍单独适用于各索引。即，如果您在全局段落级别将 maxGlobalRawDataSizeMB 设为 5000MB，那么 indexA 的最大值为 5000MB，indexB 最大值为 5000MB，依此类推。

此设置默认为 0，表示不限制索引上原始数据量。

maxGlobalRawDataSizeMB 设置仅可用于运行 7.3.0 或更高版本的索引器。

frozenTimePeriodInSecs

此设置和非 SmartStore 索引使用的设置一样。此设置会指定数据桶到达配置时间时冻结。默认值为 188697600 秒，或者约为 6 年。

有关此设置的详细信息，请参阅“当数据时间太久时将其冻结”。

索引器群集上冻结数据桶的流程。

索引器群集上冻结 SmartStore 索引的数据桶的过程按以下方式进行：

1. 管理器节点默认会每 15 分钟在所有对等节点上运行一次搜索，以识别任何需要冻结的数据桶。

通过管理器节点上 server.conf 中的 remote_storage_retention_period 设置控制搜索周期。

2. 对于每个要冻结的数据桶，管理器节点会随机将任务分配给具有数据桶本地副本的一个对等节点，即分配给索引的 .bucketManifest 文件中具有数据桶元数据的对等节点。
3. 对于每个要冻结的数据桶：
   1. 指定的对等节点会检查数据桶是否在远程存储上。只有数据桶存在于远程存储中，才会继续冻结该数据桶。否则，对等节点会跳过数据桶。

      温数据桶不在远程存储中的一个最可能的原始是，数据桶是否刚从热数据桶滚动为温数据桶。

   2. 指定的数据桶对等节点采取的下一步措施会有区别，具体取决于在删除数据桶之前是否配置群集对等节点以对冻结数据桶进行归档：
      - 如果配置群集对等节点对数据桶进行归档，那么如果数据桶不在本地缓存中，指定的对等节点会从远程存储中提取数据桶。然后会对数据桶进行归档并删除其本地副本。
      - 如果配置对等节点以在没有归档的情况下删除冻结数据桶，那么指定的对等节点不会提取数据桶。只会删除数据桶的本地副本。

      有关如何配置归档的信息，请参阅“归档索引的数据”。
   3. 指定的对等节点会删除远程存储中的数据桶。
   4. 从远程存储中删除后，指定的对等节点会通知管理器节点。
   5. 管理器节点会通知其他具有该数据桶本地副本的对等节点删除副本。

在本上下文中，术语 "local bucket copy"（本地数据桶副本）指对等节点上数据桶相关的所有信息，可以选择包括实际的数据桶副本。当对等节点在冻结过程中删除其本地数据桶副本时，它会从索引的 .bucketManifest 文件中删除数据桶的元数据，以及其缓存中数据桶的任何副本。如果缓存中没有副本，那么会删除该数据桶的空白目录。

独立索引器上的冻结数据桶流程

独立索引器上冻结 SmartStore 索引的数据桶的过程按以下方式进行：

1. 索引器默认每 60 秒检查一次，以识别需要冻结的任何数据桶。

   通过 indexes.conf 中的 rotatePeriodInSecs 设置控制服务周期。

2. 对于每个要冻结的数据桶：
   1. 索引器会检查数据桶是否在远程存储上。只有数据桶存在于远程存储中，才会继续冻结该数据桶。否则，会跳过数据桶。

      温数据桶不在远程存储中的一个最可能的原始是，数据桶是否刚从热数据桶滚动为温数据桶。

   2. 索引器采取的下一步措施会有区别，具体取决于在删除数据桶之前是否配置索引器以对冻结数据桶进行归档：
      - 如果配置索引器对数据桶进行归档，那么如果数据桶不在本地缓存中，索引器会从远程存储中提取数据桶。然后会对数据桶进行归档并删除其本地副本。
      - 如果配置索引器以在没有归档的情况下删除冻结数据桶，那么索引器不会提取数据桶。只会删除数据桶的本地副本。

      有关如何配置归档的信息，请参阅“归档索引的数据”。
   3. 索引器会删除远程存储中的数据桶。

在本上下文中，术语 "local bucket copy"（本地数据桶副本）指索引器上数据桶相关的所有信息，可以选择包括实际的数据桶副本。当索引器在冻结过程中删除其本地数据桶副本时，它会从索引的 .bucketManifest 文件中删除数据桶的元数据，以及其缓存中数据桶的任何副本。如果缓存中没有副本，那么会删除该数据桶的空白目录。

解冻数据和 SmartStore

您无法将归档的数据桶解冻到 SmartStore 索引中，即使在冻结之前，数据桶是 SmartStore 索引的一部分。

但是，您可以将数据桶解冻到非 SmartStore 索引的解冻目录中。将数据桶解冻为非 SmartStore 索引时，必须确保其数据桶 ID 在该索引中是唯一的。

如果您打算频繁解冻数据桶，您可能想要新建一组名称上与 SmartStore 索引并行的非 SmartStore 索引。例如，"nonS2_main"。

有关数据桶 ID 的信息，请参阅“数据桶名称”。

关于解冻数据桶的信息，请参阅“解冻 4.2+ 归档”。

添加 SmartStore 索引

添加 SmartStore 索引的过程和添加非 SmartStore 索引的过程类似。您可以在 indexes.conf 中新建索引段落，并配置路径信息和其他设置。

您不能通过 Splunk Web 添加、编辑或删除 SmartStore 索引。

前提条件

阅读：

- 创建自定义索引
- 配置 SmartStore

通过直接编辑 indexes.conf 添加 SmartStore 索引

您可以通过直接编辑 indexes.conf 添加 SmartStore 索引。您可以将此方法用于索引器群集和独立索引器。

和所有的 indexes.conf 设置一样，将索引添加到索引器群集上的对等节点时请使用配置软件包方法。

请注意以下事项：

- 通常会在全局级别而不是单个索引级别配置 indexes.conf 中的 SmartStore 相关的设置。在此情况下，添加新索引时不需要指定这些设置。但是，如果您正在使用多个远程卷，或您混合使用 SmartStore 和非 SmartStore 索引，则必须在索引级别指定 SmartStore 设置。
- 如果在全局级别配置 SmartStore 设置，那么新的索引段落通常只需要 homePath、thawedPath 和 coldPath 设置。即使 SmartStore 不使用这些设置，您也必须指定 coldPath 和 thawedPath 值。
- 使用 maxDataSize = auto（默认值为 750MB）。
- 如果是索引器群集，您必须指定 repFactor = auto。
- 请参阅 indexes.conf 中的 SmartStore 设置获取 SmartStore 相关设置列表。

例如，假设您已在全局级别设置所有 SmartStore 相关配置。那么，要配置名为 "cs_index" 的新索引，只需将以下段落添加到 indexes.conf 中：

```
[cs_index]
homePath = $SPLUNK_DB/cs_index/db
thawedPath = $SPLUNK_DB/cs_index/thaweddb
coldPath = $SPLUNK_DB/cs_index/colddb
```

SmartStore 故障排除

SmartStore 和 Splunk Enterprise 的其他功能一样，可提供多种工具供您对部署进行故障排除。

- 监视控制台
- 日志文件
- CLI 命令
- REST 端点

本主题将讨论用于 SmartStore 故障排除的各个工具。此外，本主题还介绍了一些常见的 SmartStore 问题和可能的原因。

用监视控制台进行故障排除

您可以使用监视控制台监视部署的大多数方面。本节介绍可用来深入了解 SmartStore 活动和性能的控制台仪表板。

监视控制台的主要文档位于*《监视 Splunk Enterprise 手册》*内。

多个仪表板监控 SmartStore 状态。仪表板的使用范围或者是单个实例，或者是整个部署。在**索引**菜单和 **SmartStore** 子菜单下找到仪表板：

- SmartStore Activity: 实例
- SmartStore Activity: 部署
- SmartStore Cache Performance: 实例
- SmartStore Cache Performance: 部署

SmartStore Activity 仪表板

SmartStore Activity 仪表板提供远程存储相关的活动信息，如：

- 远程存储连接
- 数据桶上传/下载活动
- 数据桶上传/下载故障计数

SmartStore Activity 仪表板还包括一些复选框，如果您当前正在执行数据迁移或启动，可以从中选择显示进程。

SmartStore Cache Performance 仪表板

SmartStore Cache Performance 仪表板提供本地缓存相关信息，如：

- 影响缓存逐出的 server.conf 设置的值
- 数据桶逐出率
- 用于从远程存储下载数据桶的搜索时间部分
- 缓存成功率
- 重复数据桶下载

有关更多信息，请查看仪表板本身。此外，请参阅*《监视 Splunk Enterprise》*中的“索引：索引和卷”。

使用日志文件进行故障排除

几个日志文件可深入检查 SmartStore 操作的状态。

splunkd.log。检查这些日志渠道：

- S3Client。与 S3 的通信。
- GCSClient。与 GCS 的通信。
- StorageInterface. 外部存储活动（比 S3Client 或 GCSClient 的级别更高）。
- CacheManager. 缓存管理器组件的活动。
- CacheManagerHandler。缓存管理器 REST 端点活动（服务器和客户端）。
- KeyProviderManager。与密钥提供程序设置和配置相关的错误。当系统在远程存储上加密了数据时，将使用密钥提供程序。

search.log 。检查这些日志渠道：

- CacheManagerHandler。使用缓存管理器 REST 端点活动进行数据桶操作。
- S2BucketCache。搜索时数据桶管理（打开、关闭等）。
- BatchSearch, CursoredSearch, IndexScopedSearch, ISearchOperator。数据桶相关的搜索活动。

audit.log

- 包含数据桶操作信息，如上载、下载、逐出等。

metrics.log

- 包含涉及外部存储操作的指标。

splunkd_access.log

- 包含针对缓存管理器 REST 端点的搜索流程活动的跟踪。

使用 dbinspect 获取有关 SmartStore 数据桶的信息

您可以在搜索时结合使用 dbinspect 命令，以获取有关 SmartStore 数据桶的信息。必须了解 cached 参数对 SmartStore 数据桶搜索的影响，这点很重要。

如果将 cached 参数设置为 "t"，则 dbinspect 会从数据桶的清单中获取其统计信息。如果设置为 "f"，则 dbinspect 会检查数据桶本身。

对于 SmartStore 索引，cached 的默认值为 "t"；对于非 SmartStore 索引，其默认值为 "f"。不要更改默认值。

尽管 cached 参数与 SmartStore 所用的“缓存”一词无关，但对于 SmartStore 数据桶，cached 参数的影响非常重要，因为带 cached=f 的 dbinspect 会检查数据桶的索引器本地副本，而带 cached=t 的 dbinspect 会检查数据桶清单，其中包含有关驻留在远程存储中的数据桶规范版本的信息。

为了解这种区别的重要性，让我们来看一下 sizeOnDiskMB，它是 dbinspect 返回的其中一个字段。如果索引器的本地缓存中不再有 SmartStore 数据桶的副本，则数据桶目录为空，因此其大小为 0。因此，如果 cached=f，则 dbinspect 会检查本地数据桶以确定 sizeOnDiskMB 的值。由于数据桶的本地版本仅包含空目录，因此即使远程存储中有完整的数据桶，dbinspect 也会针对此数据桶返回大小为 0。同样，数据桶的缓存副本可能仅包含完整数据桶中文件的子集，并且 cached=f，则只会返回该子集的大小，而非数据桶的整个大小。

但是，可以通过数据桶的清单获得数据桶的真实大小（即，远程存储中数据桶的规范副本的大小）。因此，如果 cached=t，则索引器从远程存储中的副本中提取数据桶清单，然后 dbinspect 会使用该清单中的统计信息，从而返回数据桶的真实大小。

有关 dbinspect 的更多信息，请参阅*《搜索参考》*中的 dbinspect。

测试远程存储连接性

远程存储连接性问题是一个常见问题。网络或权限问题可能会导致连接性问题。使用 `splunkd cmd rfs` 命令测试远程存储连接性。本部分介绍了此命令的一些用法。

列出了远程存储上 "foobar" 索引的内容：

```
splunk cmd splunkd rfs ls index:foobar
```

列出了远程存储上指定数据桶的内容：

```
splunk cmd splunkd rfs ls bucket:foo~737~B1CE2AB0-CE4A-4697-83F2-1C5DBFB6485A
```

测试从远程存储获取文件：

```
splunk cmd splunkd rfs getF bucket:foo~737~B1CE2AB0-CE4A-4697-83F2-1C5DBFB6485A/guidSplunk-B1CE2AB0-CE4A-4697-83F2-1C5DBFB6485A/Hosts.data  /tmp/foo/
```

用 REST 搜索进行故障排除

使用以下搜索获取正在搜索的数据桶列表：

```
| rest /services/admin/cacheman search=cm:bucket.ref_count>0
```

搜索打开数据桶时，`ref_count` 值会增加 1；当搜索关闭数据桶时，该值会减少 1。

使用以下搜索获取尚未上载到远程存储（即未在远程存储上处于“稳定”状态）的数据桶列表：

```
| rest /services/admin/cacheman search=cm:bucket.stable=0
```

常见问题

搜索运行缓慢或遇到问题

运行缓慢或出现问题的搜索通常由以下问题造成：

- 远程存储的性能问题。
- 缓存管理器逐出数据桶过于频繁。
- 冷缓存问题。当参与搜索的索引器群集对等节点没有某些需要的数据桶的本地副本时，会出现冷缓存，因此，必须从远程存储下载数据桶。冷缓存可能由管理器节点将主要数据桶副本重新分配到不同的对等节点产生。

搜索出错

搜索相关错误消息 "Failed to localize fileSet='....' for bid='...'.Results will be incomplete." 表示下载指定数据桶时的错误条件。

详细信息，请检查出现错误的索引器上的 `splunkd.log`。

磁盘满载问题

磁盘满载相关消息表示缓存管理器无法逐出足量的数据桶。这可能是由以下几种原因造成：

- 搜索负荷超出本地存储。例如，由至少一个搜索进程打开的数据桶可能占据完整的缓存。搜索结束后，这个问题也不复存在。
- 缓存管理器问题。如果搜索完成后问题仍然存在，那么可能是缓存管理器出现问题。请检查出现错误的索引器上的 `splunkd.log`。

SmartStore 如何工作

SmartStore 缓存管理器

每个索引器都包含一个缓存管理器，用于管理本地存储中的 SmartStore 数据。缓存管理器尝试在本地缓存中维护可能参与未来搜索的任何数据。通过缓存搜索工作集，缓存管理器最大限度地减少了从远程存储中下载数据所导致的搜索延迟的可能性。

每个索引器的缓存管理器会在其他索引器上独立操作。

缓存管理器可执行以下功能：

- 当数据桶从热数据桶滚动到温数据桶时，管理器会将数据桶从本地存储复制到远程存储。
- 当搜索需要文件时，会将数据桶文件从远程存储提取到本地存储缓存中。
- 当未来搜索可能不再需要文件时，会将文件逐出本地存储缓存。

缓存管理器如何处理滚动到温数据桶的热数据桶

当热数据桶滚动到温数据桶时，缓存管理器会将数据桶副本上载到远程存储中。上载到远程存储之后，数据桶符合被从本地缓存逐出的条件。请参阅“索引如何在 SmartStore 中工作”了解这一过程的详细信息。

数据桶上载到远程存储之后，数据桶中的文件不会更改，但是之后可向其中添加少数文件。例如，缓存管理器可上载报表加速摘要或删除日志，因为这些是在索引器中新建的。

缓存管理器如何提取数据桶

缓存管理器会从远程存储中提取数据桶文件以满足搜索请求。

缓存管理器从远程存储中提取文件之后，会将文件复制到本地缓存中。文件的主要副本会保留在远程存储中。

缓存管理器并不总是从远程存储中提取整个数据桶。相反，如果搜索可能不需要整个数据桶，管理器可以提取单个数据桶文件。例如，先获取 bloomfilter 文件，在某些情况下，缓存管理器不必提取数据桶的剩余文件。同样地，某些搜索（如 metadata 和 tstats）不需要原始数据 journal.gz 文件。其他搜索（如 dbinspect 和 eventcount）不需要任何数据桶内容。

缓存管理器尝试在搜索期间预提取数据桶，以便在搜索准备好访问数据桶时，每个数据桶都已经是本地数据桶。在预提取数据桶时，缓存管理器使用一种启发式算法，根据其他数据桶中的 bloomfilter 和 tsidx 文件如何消除结果而进行调整。因此，缓存管理器可以预提取整个数据桶或仅获取数据桶内的单个文件。例如，在搜索过程中，如果搜索多个之前的数据桶不需要 tsidx 文件，则缓存管理器不会预提取 tsidx 文件，因为它继续处理其他数据桶。

在预获取时，缓存管理器会记录搜索进程对每个数据桶所花的时间，并且会以允许不打断继续搜索的速度预提取数据桶，而无需等待数据桶提取完成。缓存管理器还会调整速度以避免预提取过多的数据桶。

缓存管理器如何逐出数据桶

缓存管理器会通过保留哪些很可能会参与未来搜索和逐出其他搜索的数据桶和文件，尝试尽量减少使用本地存储。缓存管理器逐出数据桶之后，会删除驻留在缓存上的数据桶副本。因为缓存的数据桶为主节点副本驻留在远程存储中的数据桶的本地副本，逐出程序不会导致数据丢失。

将数据桶从缓存中逐出之后，其目录仍会驻留在缓存中，但是目录是空白的。数据桶索引的 .bucketManifest 文件还会保留数据桶的元数据。

缓存管理器不一定会逐出数据桶中的所有文件。这非常适合逐出较大的文件，例如原始数据文件和 tsidx 文件，在缓存中留下较小的文件，如 bloomfilter 和 metadata。这样缓存管理器可降低从远程存储中提取较大文件的需求。例如，在搜索开始时检查数据桶的 bloomfilter 文件有时可以消除搜索数据桶数据的需求，因此无需提取数据桶的原始数据日志和 tsidx 文件。可通过缓存管理器新近程度设置来配置此行为。请参阅“根据新近程度设置缓存保留期”。

缓存管理器会根据可配置的策略操作。默认策略是 "lru"，该策略告诉缓存管理器逐出最近很少使用的数据桶。要了解一组可用逐出策略，请参阅“设置逐出策略”。

配置缓存管理器

请参阅“配置 SmartStore 缓存管理器”。

索引如何在 SmartStore 中工作

索引器处理 SmartStore 索引中的数据桶和处理非 SmartStore 索引中的数据桶的方式不同。

数据桶阶段和 SmartStore

索引器会将非 SmartStore 索引中的数据桶保留为以下状态：

- 热数据桶
- 温数据桶
- 冷数据桶

索引器会将 SmartStore 索引中的数据桶保留为以下状态：

- 热数据桶
- 温数据桶

SmartStore 索引的热数据桶会驻留在本地存储中，和非 SmartStore 索引一样。温数据桶会驻留在远程存储中，尽管这些数据桶的副本可能也会临时驻留在本地存储中。

不存在冷数据桶的概念，因为不需要再区分温数据桶和冷数据桶。在非 SmartStore 索引中，冷数据桶状态仍然存在，可以用于识别可安全移动到某种更廉价的存储中的旧数据桶，因为数据桶的搜索频率通常会随着时长的增长而降低。但是对于 SmartStore 索引，温数据桶仍存在于廉价存储上，因此没有理由随着时长的增加将这些数据桶移动到其他类型的存储中。

温数据桶直接滚动为冻结数据桶。

事实上，冷数据桶可存在于 SmartStore 索引中，但是只在限定情形下。尤其是，如果您将某个索引从非 SmartStore 迁移到 SmartStore，迁移后任何已迁移的冷数据桶将使用现有的冷路径作为其缓存位置。

SmartStore 索引中的冷数据桶各方面功能和温数据桶相同。缓存管理器可用和管理温数据桶相同的方式管理已迁移的冷数据桶。惟一的区别在于冷数据桶在必要时将被提取到冷路径位置而不是主路径位置。

索引过程

SmartStore 和非 SmartStore 索引的索引过程是一样的。

索引器为传入数据建立索引并将数据写入本地存储的热数据桶中。如果是索引器群集，来源对等节点将使热数据桶数据流入目标对等节点以满足复制因子。

但是，当数据桶变为温数据桶时，SmartStore 进程会和非 SmartStore 不同。

温数据桶处理

热数据桶变为温数据桶时，索引器处理 SmartStore 索引的方式与非 SmartStore 索引不同。

SmartStore 索引中的数据桶变为温数据桶时，数据桶将被复制到远程存储中。

滚动的数据桶不会立即从索引器的本地存储中移除。相反，它会在本地缓存，直到它被逐出以响应缓存管理器的逐出策略。由于搜索往往在最近的数据中最频繁地发生，因此此过程有助于最小化需要从远程存储检索以实现搜索请求的数据桶数量。

在缓存管理器最终从索引器的本地存储中删除数据桶之后，索引器仍然在索引的 `.bucketManifest` 文件中保留该数据桶的元数据信息。此外，索引器会为数据桶保留空白目录。

注意：在某些情况下，当缓存管理器逐出数据桶时，它会保留 `bloomfilter` 文件以及其他一些小文件。也就是说，它会删除数据桶的 `rawdata` 日志和 `tsidx` 文件，但会把小文件暂时保留在原处。可通过缓存管理器新近程度设置来配置此行为。请参阅“根据新近程度设置缓存保留期”。

如果是索引器群集，当数据桶滚动为温数据桶时，来源对等节点会将数据桶上载到远程存储中。来源对等节点会继续将数据桶副本保留在本地缓存中，直到适当的时候缓存管理器将副本逐出。

成功上载数据桶之后，来源对等节点会将消息发送到数据桶目标对等节点中，通知该等对等节点数据桶已上载至远程存储中。目标对等节点（如来源对等节点）会继续将数据桶副本保留在本地缓存中，直到适当的时候缓存管理器将副本逐出。

上载过程中，如果目标对等节点未在五分钟内收到来源对等节点的任何通知，目标对等节点将查询远程存储以了解是否已上载数据桶。如果来源对等节点尚未上载数据桶，那么其中一个目标对等节点将上载。

当数据桶副本确实从对等节点的本地缓存中逐出时，对等节点会保留数据桶的元数据，这样群集可以元数据形式拥有足够多的数据桶副本，以匹配复制因子。

除保留数据桶的元数据信息外，来源对等节点还会继续保留数据桶的主要指定条件。当需要搜索数据桶时，具有主要指定条件的对等节点会提取远程存储的数据桶。

SmartStore 如何处理报表和数据模型加速摘要

启用 SmartStore 的索引器会以和启用非 SmartStore 的索引器类似的方式处理摘要。在索引器群集中，摘要在对等节点上新建，该对等节点是关联的一个或多个数据桶的主要节点。然后对等节点会将摘要上载至远程存储。当对等节点需要摘要时，其缓存管理器会从远程存储中捕获摘要。

摘要复制是不必要的，因此不支持摘要复制，因为上载的摘要可用于所有对等节点。

使用 SmartStore 时，设置 summaryHomePath 和 tstatsHomePath 必须保持未设置。请参阅“indexes.conf 中和 SmartStore 不兼容或有其他方面限制的设置”。

有关索引器集群中的报表和数据模型加速摘要的详细信息，请参见“索引器群集如何处理报表和数据模型加速摘要”。关于报表和数据模型加速的常规信息，请参阅《知识管理器手册》中的“管理报表加速”和“加速数据模型”。

数据桶冻结和 SmartStore

请参阅“为 SmartStore 索引配置数据保留”。

搜索如何在 SmartStore 中工作

SmartStore 针对大多数 Splunk 平台搜索所共有的特定特性进行了优化。尤其是大部分搜索具有以下特征：

- 针对近期数据进行搜索。97% 的搜索都会回溯到过去 24 小时或更短时间。
- 具有空间和时间局限性。如果搜索在特定时间或特定日志中发现事件，那么其他搜索可能会在相似的时间范围内或该日志中查找事件。

缓存管理器支持最近创建的存储桶和访问的存储桶，试图确保搜索可能需要的大多数数据存在于本地缓存中。类似地，缓存管理器倾向于逐出可能不经常参与搜索的数据桶。

对于 SmartStore 和非 SmartStore 索引，大多数搜索基础都是相同的。索引器会搜索数据桶以响应搜索头请求。搜索数据桶的过程始终发生在本地存储中。作为任何搜索的第一步，索引器会编译其需要搜索的数据桶列表。

要了解搜索如何在 SmartStore 索引上工作以及与非 SmartStore 索引上的搜索有何差别，有必要区分这些数据桶状态：

- 热数据桶
- 驻留在本地缓存中的温数据桶
- 不是驻留在本地缓存中的温数据桶

热数据桶始终驻留在本地存储中，无论数据桶是 SmartStore 索引的一部分还是非 SmartStore 索引的一部分。因此，两种环境中热数据桶工作方式相同。

但是，温数据桶搜索过程会不同。

当索引器需要在热数据桶中搜索 SmartStore 索引时，它会先“打开”该桶。随后，数据桶搜索完成后，会“关闭”该数据桶。“打开”和“关闭”的指定对于高速缓存逐出过程的正确运行是必要的。当数据桶打开时，索引器的缓存管理器知道数据桶正在参与搜索，因此不符合逐出条件。数据桶关闭后符合逐出条件。

打开温数据桶之后，索引器会决定数据桶目前是否驻留在本地缓存中。下一个处理阶段取决于存储桶当前是否在缓存中：

- 如果数据桶驻留在本地缓存中，索引器会正常搜索数据桶。搜索完数据桶并关闭后，该数据桶就符合逐出条件。
- 如果数据桶不驻留在本地缓存中，缓存管理器必须先从远程存储中提取副本并放入本地缓存中。之后索引器会正常搜索数据桶。搜索完数据桶并关闭后，该数据桶就符合逐出条件。

注意：在某些情况下，缓存管理器只提取特定的数据桶文件，而不是整个数据桶。请参阅“缓存管理器如何提取数据桶”。

如果是索引器群集，主要数据桶副本功能和非 SmartStore 索引类似。对于热数据桶，只有数据桶的主要副本会参与搜索。对于温数据桶，只有具有该数据桶主要指定条件的对等节点才能提取数据桶或访问缓存副本。

因为数据桶的主要指定条件很少在对等节点之间切换，缓存中的温数据桶通常会驻留在具有该数据桶主要指定条件的对等节点上。同一个温数据桶的副本驻留在多个对等节点的缓存中是不常见的，或者数据桶的主要指定条件驻留在没有本地副本的对等节点上，而另一个对等节点已经具有该桶的本地副本也是不常见的。

索引器群集操作和 SmartStore

索引器集群在某些基本方面对 SmartStore 索引的处理方式与非 SmartStore 索引不同：

- SmartStore 温数据桶的高可用性和灾难恢复责任从群集转移到远程存储服务。这种转换具有重要优势，即使群集丢失了一组数量上等于或超过复制因子的对等节点，也完全可以恢复温数据桶数据。
- 复制因子对 SmartStore 温数据桶的影响和对非 SmartStore 温数据桶的影响不同。特别是，群集会使用复制因子决定保

留 SmartStore 温数据桶元数据的副本数量。群集不会尝试自己保留温数据桶的多个副本。如果是温数据桶修复，群集只需要复制数据桶元数据，而不是数据桶目录的全部内容。

复制因子和复制的数据桶副本

群集会用和在非 SmartStore 索引中处理热数据桶相同的方式处理 SmartStore 索引中的热数据桶。会跨对等节点的对等因子数复制本地存储中的热数据桶。

当 SmartStore 索引中的数据桶滚动到温数据桶状态，并移动到远程存储之后，远程存储服务会保持该数据桶的高可用性。复制因子对远程存储如何实现该目标没有影响。

当数据桶滚动到温数据桶状态并上载到远程存储时，对等节点不再尝试保留数据桶本地副本的复制因子数。对等节点会将数据桶的副本保留在本地缓存中一段时间，具体由缓存逐出策略决定。

然而，即使数据桶副本不再存储在本地，复制因子仍会控制群集为该数据桶存储的元数据。与复制因子数量相等的对等节点在其 .bucketManifest 文件中保留有关数据桶的元数据信息。如果数据桶副本目前没有驻留在本地缓存中，这些对等节点还会保留数据桶的空白目录。

例如，如果群集复制因子为 3，则三个对等节点会继续维持每个数据桶的元数据信息、填充的或空的目录。

通过在对等节点的复制因子数上保留每个桶的元数据，在任何临时对等节点故障的情况下，群集简化了在需要时提取数据桶的过程。

和非 SmartStore 索引不同，如果群集丢失等于或超过复制因子数量的对等节点，群集可以恢复 SmartStore 索引中的大部分数据。在此情况下，群集可恢复所有的 SmartStore 温数据桶，因为这些数据桶存储在远程存储中，因此对等节点故障不会影响这些数据。群集可能会丢失部分 SmartStore 热数据桶，因为这些数据桶是在本地存储。

例如，在复制因子为 3 的群集中，如果有三个或更多对等节点同时脱机，那么索引会丢失非 SmartStore 索引中的热数据和温数据。但是，同一个群集可能会丢失任意数量的对等节点，甚至临时丢失所有对等节点，但仍然不会丢失任何 SmartStore 温数据，因为该数据驻留在远程存储上。

搜索因子、可搜索副本和主要副本

搜索因子对 SmartStore 索引中热数据桶的影响和对非 SmartStore 索引中热数据桶的影响是一样的。也就是说，搜索因子确定包含 tsidx 文件的每个复制数据桶的副本数量，并因此可搜索。

对于 SmartStore 温数据桶，搜索因子没有实际含义。远程存储保存每个数据桶的主副本，并且该副本始终包含一组 tsidx 文件，因此根据定义是可搜索的。当响应搜索请求，对等节点的缓存管理器会将数据桶的副本提取到节点的本地缓存时，该副本的可搜索程度与其针对特定搜索的可搜索程度相同。如缓存管理器如何提取数据桶中所述，缓存管理器尝试仅下载特定搜索所需的数据桶文件。因此,在某些情况下,缓存管理器可能不会下载 tsidx 文件。

主要标记与 SmartStore 索引的结合使用的方式与非 SmartStore 索引的方式一样。每个数据桶只有一个对等节点，该节点具有该数据桶的主要标记。对于每次搜索，具有特定温数据桶的主要标记的对等节点是搜索该数据桶的对等节点，必要时首先从远程存储提取该数据桶的副本

索引器群集如何处理由对等节点故障产生的 SmartStore 数据桶修复

如果对等节点故障，索引器群集可使用和非 SmartStore 索引基本相同的方式处理 SmartStore 索引的数据桶修复，所用方式略有不同。有关对等节点故障和随后的数据桶修复活动的一般讨论，请参阅“对等节点发生故障时会发生什么”。

这部分介绍使用 SmartStore 时出现的数据桶修复的不同。SmartStore 的一个优势在于丢失等于或超出复制因子数的对等节点不会导致丢失温数据桶数据。此外，温数据桶修复会比 SmartStore 索引更快。

使用 SmartStore 进行数据桶修复

如果是热数据桶，SmartStore 索引的数据桶修复和非 SmartStore 索引的数据桶修复方式相同。

如果是温数据桶，SmartStore 索引的数据桶修复比非 SmartStore 索引快得多。之所以出现这种优势，是因为 SmartStore 数据桶修复只需要更新每个对等节点上的 .bucketManifest 文件，而无需自行传输数据桶。换句话说，数据桶修复只复制数据桶元数据。

每个对等节点仍保留其各自索引的 .bucketManifest 文件。该文件包含对等节点维护的每个数据桶副本的元数据。将数据桶从来源对等节点复制到目标对等节点时，目标对等节点会将元数据添加到其数据桶副本的 .bucketManifest 文件中。

在 SmartStore 修复期间不会复制数据桶本身，因为每个数据桶的主要副本仍然存在于远程存储上，并且仅在搜索需要时才下载到对等节点的本地存储中。

在 SmartStore 复制期间，目标对等节点还会为每个 .bucketManifest 文件中有元数据的温数据桶创建空白目录。

和非 SmartStore 索引一样，SmartStore 数据桶修复过程还会确保只有一个对等节点具有每个数据桶的主要标记。

故障的对等节点数量小于复制因子数时的修复

对于 SmartStore 和非 SmartStore 索引，群集通过修复过程可完全恢复其有效的完整状态。SmartStore 温数据桶的修复过程只需要复制群集内部的元数据，因此修复速度更快。

故障的对等节点数量等于或超过复制因子数时的修复

和非 SmartStore 索引相反，即使故障的对等节点数量等于或超出复制因子，群集也可以恢复其 SmartStore 索引的全部温数据桶数据。尽管和非 SmartStore 索引一样，但是也可能会丢失一些和热数据桶相关的数据。

群集只需要恢复丢失的温数据桶元数据，因为温数据桶本身仍会驻留在远程存储中。

要恢复丢失的温数据桶元数据，管理器节点会使用其所有数据桶 ID 的完整列表。将这些 ID 和对等节点上 .bucketManifest 文件集中的 ID 进行比较，查找存在于列表中但不存在于任何 .bucketManifest 文件中的 ID。对于所有此类数据桶 ID，管理器节点会分配对等节点以查询远程存储，获取填充 .bucketManifest 文件中数据桶元数据所需的信息。进行其他修复以满足元数据的复制因子要求并分配主要标记。

如果在修复过程中管理器节点出现故障，那么温数据桶元数据仍可恢复。当群集重新获取管理器节点时，管理器节点会启动启动流程以恢复所有温数据桶元数据。有关启动的信息，请参阅“启动 SmartStore 索引”。

索引器和索引器群集故障排除

非群集数据桶问题

本节将介绍如何处理独立于群集化存在的数据桶的各种问题。

重新构建所有的数据桶

索引器通常可以在无人介入的情况下完成崩溃的恢复。如果索引器意外关闭，某些最近接收的数据可能将变成不可搜索。您重新启动索引器时，它会在后台自动运行 fsck 命令。该命令将诊断数据桶的运行状况，并会在需要时重新构建搜索数据。

警告：需要您手动运行 fsck 的机率不大。这样很好，因为如果要手动运行该命令，您必须停止索引器，并且在索引庞大的情况下，该命令需要花费数小时才能完成。在此过程中，您的数据是无法访问的。但是，如果 Splunk 支持引导您运行该命令，本节的后文将介绍如何执行此操作。

要手动运行 fsck，首先必须停止索引器。然后，针对受影响的数据桶运行 fsck。要针对所有索引中的数据桶运行 fsck，请使用以下命令：

```
splunk fsck repair --all-buckets-all-indexes
```

这会重新构建所有索引中所有类型的数据桶（热/温/冷）。

若要只重新构建单个索引中所有的数据桶，请使用此版本的命令：

```
splunk fsck repair --all-buckets-one-index --index-name=<your_index>
```

注意：fsck 命令只重新构建由 4.2 版或更高版本 Splunk Enterprise 创建的数据桶。

fsck repair 命令运行可能需要数小时，具体取决于索引的大小。如果您可以确定您只需重新构建几个数据桶，那么可以按照下一部分“重新构建单个数据桶”中的介绍，仅针对这几个数据桶运行 rebuild 命令。

如果只想诊断索引的状态（而不立即采取任何补救措施），可运行：

```
splunk fsck scan --all-buckets-all-indexes
```

要了解有关 fsck 命令的更多信息，包括所有可用选项的列表，输入：

```
splunk fsck --help
```

重新构建单个数据桶

如果数据桶（4.2 版及更高版本）中的索引和元数据文件由于某种原因被破坏，可以只使用元数据文件来重新构建数据桶。使用以下命令：

```
splunk rebuild <bucket directory><index-name>
```

索引器会自动删除旧索引和元数据文件，并重新进行构建。您不需要亲自删除任何文件。

注意：

- 重新构建数据桶并不计入您的许可证容量。
- 重新构建数据桶可能非常耗时。根据不同的系统因素，例如，您的硬件规格，重新构建 10 GB 数据桶可能需要花费半小时到数小时不等的时间。
- splunk rebuild 是 splunk fsck repair -one-bucket 的别名。

恢复无效的 4.2 版之前的热数据桶

热数据桶会变成无效的热（invalid_hot_<ID>）数据桶，前提是索引器检测到元数据文件（Sources.data, Hosts.data, SourceTypes.data）遭到破坏或不正确。不正确的数据通常意味着不正确的时间范围，还可能表示事件计数不正确。

索引器会忽略无效的热数据桶。数据不会添加到这种数据桶中，也无法对这种数据桶进行搜索。在确定数据桶限值（如 maxTotalDataSizeMB）时，也不会将无效的数据桶考虑在内。这表示无效的数据桶不会对通过系统的数据流造成负面影响，但这也表示它们会导致磁盘存储超过配置的最大值。

要恢复无效的热数据桶，使用 recover-metadata 命令：

1. 创建元数据文件 Sources.data, Hosts.data, SourceTypes.data 的备份副本。

2. 基于原始数据信息重新构建元数据：

```
splunk cmd recover-metadata path_to_your_hot_buckets/invalid_hot_<ID>
```

3. 成功后，按照正常的命名方式重命名该数据桶。

重新构建索引级数据桶清单

索引级数据桶清单文件是 .bucketManifest。文件包含索引中所有数据桶的列表。

通常不需要重新构建清单。如果您手动将数据桶复制到索引，这种情况下可能需要重新构建清单。

只有 Splunk 支持向您发出指示时，您才可以重新构建清单。不要自行重新构建清单。

此命令仅为 main 索引重新构建 .bucketManifest 文件：

```
splunk _internal call /data/indexes/main/rebuild-bucket-manifest
```

要为所有索引重新构建清单，请使用星号 (*) 通配符：

```
splunk _internal call /data/indexes/*/rebuild-bucket-manifest
```

数据桶复制问题

网络问题阻碍了数据桶复制

如果对等节点之间的连接出现问题，使得数据来源对等节点无法复制热数据桶到目标对等节点，则源对等节点会滚动热数据桶并启动新热数据桶。如果与目标对等节点的连接仍然存在问题，它将滚动新热数据桶等等。

为防止延长的故障导致数据来源对等节点生成大量小热数据桶，则在单个目标对等节点的可配置复制错误数量后，数据来源对等节点将停止滚动热数据桶，以响应其与该目标对等节点的连接问题。默认值是三个复制错误。然后，以下横幅消息将在管理器节点的仪表板中显示一次或多次，这取决于数据来源对等节点出现错误的数量：

```
Search peer <search peer> has the following message: Too many streaming errors to target=<target
peer>. Not rolling hot buckets on further errors to this target. (This condition might exist with
other targets too. Please check the logs.)
```

在网络问题持续存在的情况下，大部分最近的热数据桶可能没有复制因子数量的副本可用。

如果一个特定对等节点总是被其他对等节点报告为复制错误的原因，则您可以暂时将该对等节点置为手动滞留状态。一旦解决了根源问题，则将对等节点从手动滞留中移除。请参阅“将对等节点置为滞留状态”。

配置复制错误的允许数量

要调整复制错误的允许数量，您可以配置 max_replication_errors 属性（位于数据来源对等节点上的 server.conf 中）。然而，您不太可能需要更改将属性的默认值 3，因为单个网络问题导致的复制错误会组合并仅计数为一个错误。可能仍会显示“过多流化错误”消息，但是可以忽略。

注意：组合复制错误是在 6.0 版本中出现的变化。通过这种变化，错误数量可能不会超过默认值 3，异常情况除外。

数据来源对等节点上的复制故障的证据

复制失败证据显示在数据来源对等节点的 splunkd.log，引用了出现故障的目标对等节点。您可以搜索 "CMStreamingErrorJob" 在日志中找到相关行。例如，本 grep 命令会发现对等节点出现 15 个流化错误，其中 GUID "B3D35EF4-4BC8-4D69-89F9-3FACEDC3F46E"：

```
grep CMStreamingErrorJob ../var/log/splunk/splunkd.log* | cut -d' ' -f10 | sort |uniq -c | sort -nr
15 failingGuid=B3D35EF4-4BC8-4D69-89F9-3FACEDC3F46E
```

无法禁用和重新启用对等节点

当把索引器作为对等节点禁用时，曾在节点（当时它已被禁用）上的热数据桶会滚动到温数据桶，并以单独的数据桶命名约定命名。如果您之后重新启用对等节点，会产生一个问题，因为管理器节点记住了那些群集中的数据桶并期望他们依据群集数据桶的约定命名，而其实他们依据独立数据桶的约定命名。因为此命名不一致，对等节点不能重新加入群集。

要解决这个问题，必须清理数据桶，或重新启用之前，删除节点上单独的数据桶。

多站点群集不满足其复制或搜索因子

该症状是出现一条消息，内容是多站点群集无法满足其复制或搜索因子。例如，这条消息会出现在管理器节点仪表板上。在启动多站点群集后会立刻发生这种情况。

将单个站点 `replication_factor` 和 `search_factor` 的属性值与每个站点上的对等节点数相比较。（如果您没有明确地设置单个站点的复制和搜索因子，那么它们的默认值分别为 3 和 2。）这些属性值不得超过任意站点上的对等节点数量。如果有属性值超过了最小站点上的对等节点数量，请将该属性值更改为最小站点上的对等节点数量。例如，若最小站点上的对等节点数量为 2，而您正在使用的默认值分别为 `replication_factor=3` 和 `search_factor=2`，则您必须将 `replication_factor` 明确设置为 2。

将单个站点群集转换为多站点群集后会出现这种情况。如果您从一开始就将群集配置为多站点群集，在您首次启动时，不会出现这种情况。

异常的数据桶问题

异常的数据桶指那些无限期停留在修复状态，无任何改善的数据桶。这样的数据桶说明存在或可能会造成更大的系统问题。例如，异常的数据桶可能会阻碍群集满足其复制因子和搜索因子。

数据桶状态仪表板不仅可以让您识别异常的数据桶，还可以让您执行操作修复这些数据桶。具体而言，您可以：

- 获取数据桶的详细信息。
- 把数据桶从热滚动到温。
- 重新同步对等节点和管理器节点之间数据桶副本的状态。
- 删除单个对等节点上的数据桶副本或删除所有对等节点上的所有数据桶副本。

在数据桶上执行上述操作前请先咨询 Splunk 支持。如果在一知半解的情况下执行了其中的某些操作，可能会造成更多的系统问题，甚至造成不可撤销的数据丢失。

识别异常的数据桶

如需识别异常的数据桶并针对它们执行操作，请使用数据桶状态仪表板。

1. 从管理器节点仪表板前往数据桶状态仪表板。请参阅查看数据桶状态仪表板。
2. 单击**修复任务 - 待定**选项卡。

您可以通过修复类型和等待修复的时长过滤待定数据桶的列表。如果某个数据桶等待修复的时间异常长，它可能就是问题的原因所在。

对异常的数据桶执行操作

如果数据桶在修复时卡了太长时间，您可以采取补救措施。

1. 单击您想要管理的数据桶的**操作**按钮。
2. 从可用的操作中选择一个：
   - 查看数据桶详情
   - 滚动
   - 重新同步
   - 删除副本

   弹出的窗口会引导您执行选定的操作。

在异常的数据桶上执行操作时请遵循以下顺序。

1. 查看数据桶详情
2. 滚动
3. 重新同步
4. 删除副本

仅在上一步操作未解决问题的情况下执行下一步操作。

查看数据桶详情

弹出的窗口将提供数据桶详情，例如：

- 数据桶大小
- 数据桶是否冻结
- 数据桶是否曾遭强制滚动
- 是否为独立的数据桶
- 数据桶驻留在上面的对等节点

这些详情将帮您排查引发数据桶问题的原因并找出相应的补救措施。

滚动

此操作把数据桶从热状态滚动到温状态。此操作仅对热数据桶有效。

重新同步

管理器节点上有一数据桶所有副本的信息。但在某些情况下，管理器节点提供的一特定对等节点上的副本信息可能是错误的。管理器节点和对等节点间若出现通信故障，就会发生上述信息错误的状况。

以下这些示例中的数据桶副本状态信息可能会出现对等节点和管理器节点同步失败的情况：

- 副本是否可搜索
- 副本是热还是温
- 是否为主要副本
- 副本是否存在于上述对等节点上

对等节点了解其数据桶副本的状态，因此若对等节点和管理器节点就某个数据桶副本提供了不同的状态信息，则管理器节点上的信息是错误的。

如需解决这个问题，请重新同步管理器节点上的数据桶副本状态。重新同步数据桶后，您可以指定对等节点和您需要重新同步的副本。重新同步进程将促使对等节点把其当前有关数据桶副本的信息发送给管理器节点。

删除副本

您可以在特定对等节点上删除单个数据桶副本，也可以从整个群集上删除所有的数据桶副本。

若因删除了单个副本而导致群集丧失其完整状态，群集将运行修复活动，让数据桶可以再次同时满足搜索因子和复制因子。这种情况可能会导致同一个对等节点上出现数据桶的另一个副本。但是，如果指定的数据桶冻结了，群集不会尝试任何修复活动。

在群集上对一个数据桶的所有副本执行删除操作，将导致不可撤销的数据丢失。

配置软件包问题

本主题介绍了当配置软件包从管理器节点推送到对等节点的时候可能出现的问题。

当推送一个非常大的软件包的时候软件包验证失败

如果试图推送一个非常大的配置软件包（>200MB），软件包验证可能由于各种超时而失败。要缓解这一问题，您可以调整接收管理器节点同时推送软件包的对等节点的数目。

默认情况下，管理器节点同时向所有对等节点推送软件包。`server.conf` 中的 `max_peers_to_download_bundle` 设置提供了一种方法来限制同时接收软件包的对等节点的数目。

例如，如果您设置 `max_peers_to_download_bundle = 3`，则管理器节点每次将软件包推送给三个对等节点。当一个对等节点完成下载时，管理器节点会将软件包推送给另一个对等节点，依此类推，直到所有对等节点都收到了软件包。

有关详细信息，请参阅 `server.conf` 规范。

软件包验证仍在进行中并未完成

如果管理器节点上的软件包验证进程仍在进行中，且未显示任何验证响应，您必须取消并重置软件包推送操作以转义正在进行的状态。

要取消和重置软件包推送操作，请点击管理器上的以下端点：`cluster/manager/control/default/cancel_bundle_push`。

使用 Hadoop Data Roll 归档数据

有关使用 Hadoop Data Roll 归档索引

您的 Splunk 许可证包含 Hadoop Data Roll。Hadoop Data Roll 提供了一种用户友好的方式，让您可以把温、冷和冻结的索引数据复制为已归档数据。把 Splunk 索引数据归档入 HDFS 或 S3，方便您：

- 搜索 Splunk 内不再可用的已归档数据。
- 在已归档数据桶和索引上执行搜索。
- 对已归档的数据执行批处理分析。
- 归档索引器数据，在不占用宝贵的索引器空间的情况下遵守您的数据保留政策。

Windows 不支持 Hadoop Data Roll。

设置归档

如需配置归档，您须告知 Splunk Enterprise 以下内容：

- 要归档哪个索引。
- 要把已归档数据放入 HDFS 或 S3 中的哪里。
- 多久后应把数据桶复制到 HDFS 中的归档。

有两种方法可以配置上述信息：

- 通过编辑 `indexes.conf`
- 在 Splunk Web 中

系统要求

确保您可以访问至少一个 Hadoop 群集（其中含数据）而且可以在群集中的数据上运行 MapReduce 任务。

确保您安装了 Java Development Kit (JDK) 1.6 及以上版本。不过，为了获取最佳效果，请升级至 JDK 1.6 以上的版本。以下分发和版本已使用 JDK 1.8 进行认证。

以下几种 Hadoop 分布和版本支持 Hadoop Data Roll：

- Apache Hadoop 3.2.1
- Open Apache 3.1.2
- Cloudera Distribution including Apache Hadoop v6.3
- Hortonworks Data Platform (HDP) 3.1.4
- MapR 6.1

Hadoop 节点上需要配置什么内容

在 Hadoop TaskTracker 节点上，您需要在 *nix 文件系统上配置一个目录，来运行符合以下要求的 Hadoop 节点：

- 1GB 的免费磁盘空间，用于存放 Splunk 副本。
- 5-10GB 的免费磁盘空间，用作临时存储。该存储空间供各搜索进程使用。

Hadoop 文件系统上需要配置什么内容

在您的 Hadoop 文件系统（HDFS 或其他）上，您需要：

- 一个位于 `jobtracker.staging.root.dir` 下的子目录（通常为 /user/），该子目录以用户帐号为名称，而 Splunk Analytics for Hadoop 在该用户帐号下于搜索头上运行。例如，若 Splunk Analytics for Hadoop 由用户 "BigDataUser" 和 `jobtracker.staging.root.dir=/user/` 启动，您需要一个用户 "BigDataUser" 可以访问的目录 `/user/HadoopAnalytics`。
- 上述目录下的子目录，可供此服务器用于中间存储，如 `/user/hadoopanalytics/server01/`

搜索已归档的索引

您可以像常规搜索一样搜索已归档的数据桶，只需把归档虚拟索引包含在您的搜索中即可。有关与储存在 Hadoop 中的索引结合使用的搜索命令，请参阅“搜索已归档的索引数据”。

例如，您可以创建一个搜索，让该搜索在 Splunk 内搜索以下内容：

- Splunk Enterprise 索引内的数据。
- 复制到 HDFS 或 S3 里的已归档数据。

搜索性能

搜索归档数据时，Splunk Enterprise 将对已归档的数据执行批处理搜索，此类搜索一般比索引数据搜索慢很多。由于 Splunk 会根据您的 indexes.conf 设置删除冷数据，已归档数据可能也会出现在 Splunk Enterprise 索引中。熟悉您的归档和 Splunk 索引器保留政策及设置十分重要。掌握这些政策和设置后，若您需要查找仍存在于 Splunk 中的特定信息，才能运行更有效的搜索。

为缩短归档数据的搜索时间，您可以设定日期来限制要搜索的数据桶。Splunk 索引器数据中的存储路径包含各数据桶最旧和最新的时间。所以，当您搜索某段时间内的数据时，Splunk 可以使用该时间信息把搜索范围缩小到相关的数据桶，避免在整个已归档的索引内进行搜索。

Hadoop Data Roll 如何工作

Hadoop Data Roll 不适用于 journalCompression 设置为 zstd 的数据桶。

在将索引配置为归档后，许多进程就将旧数据迁移到归档索引中：

1. 保存的搜索 | archivebuckets 在搜索头上一小时自动运行一次。这是 archiver 附带的自定义命令，以 Python 脚本 archivebuckets.py. 实现。

2. archivebuckets 会查询本地 REST 端点来发现哪些索引应该归档，以及在何处归档索引。

3. archivebuckets 会将 Hadoop Data Roll jars 复制到自己的应用目录，然后为要归档索引的每个提供程序启动分布式搜索。

在此步骤中使用的搜索是 | copybuckets，这是一个通过 copybuckets.py 自动执行的自定义命令。

4. 索引及其提供程序的信息将提供给搜索。

5. Splunk Enterprise 为每个索引器复制运行搜索所需的知识软件包。

6. 在索引器上，copybuckets 启动一个 Java 程序，入口点（SplunkMR 类）与用于 Splunk Analytics for Hadoop 搜索的入口点相同。

7. Splunk Enterprise 使用 stdin 将有关提供程序和索引的信息传递给 Java 进程。

8. 当 Java 进程将事件发回 Splunk Enterprise 时，它将它们写入 stdout，并且自定义的搜索命令（copybuckets.py）使用 stdout 将它们写回搜索进程。

9. Java 进程将这些操作记录到 splunk_archiver.log 文件中。

10. Java 进程检查指定索引中的所有数据桶。如果数据桶已准备就绪可以归档，则进程会确定数据桶是否已经存在于归档中。它使用提供程序信息来访问归档。

11. 如果数据桶尚未被归档，则将在归档时将数据桶复制到临时目录中。完整复制数据桶、并添加了一个收据文件后，它将被移动到归档的正确文件夹中。

12. 如果数据桶之前已归档，则已到达其归档日期的任何新数据都将复制到该数据桶中。

13. 归档的数据桶已准备就绪，可随时在 Splunk Web 中进行搜索。

关于 Hadoop Data Roll 进程

定义了两个与 Python 脚本通信的搜索命令：

```
archivebuckets -> archivebuckets.py
copybuckets -> copybuckets.py
```

Hadoop Data Roll 进程的执行使用以下进程：

进程操作	进程名称	注释
搜索头/搜索计划程序上的搜索进程	archivebuckets	这是发生在搜索头上的搜索活动，包括计划
搜索头上的 Python 进程	archivebuckets.py	

索引器上的搜索进程	copybuckets <JSON describing indexes>	这是发生在索引器上的所有搜索活动。
索引器上的 Python 进程	copybuckets.py	
索引器上的 Java 虚拟机进程	Hunk Java 代码	该进程将其他进程连接在一起，并执行以下操作： 1. 将文件写入 HDFS 2. 将信息记录到 $SPLUNK_HOME/var/log/splunk/splunk_archiver.log 3. 将事件写入 stdout，通过管道将它输送回 Splunk 搜索进程 | copybuckets <JSON describing indexes> 4. 写入 Splunk 搜索进程的信息会变成通过搜索返回的一个事件，您可以使用搜索命令 | archivebuckets forcerun=1 在 Splunk Enterprise 中查看这些事件。

各种进程如何配合

Hadoop Data Roll 搜索框架将这些进程串在一起，并按如下方式对它们进行管道传输：

1. 索引器 copybuckets.py 上的 Python 进程的 stdout 发送到索引器 archivebuckets 上的搜索进程的 stdin。

2. （属于 | copybuckets <JSON describing indexes>）的 stdout 发送到搜索头（archivebuckets.py）上的 Python 进程的 stdin。

3. 属于 archivebuckets.py 的 stdout 发送到搜索头的搜索进程搜索计划程序（archivebuckets）。

4. 搜索头上的 Python 进程 copybuckets.py 脚本将 Hadoop Data Roll Java 代码（索引器上的 Java 虚拟机进程）的 stdout 发送到索引器上的搜索进程（copybuckets <JSON describing indexes>）的 stdout。

在该进程结束时，Hadoop Data Roll Java 代码（索引器上的 JVM 进程）写入 stdout 的任何信息都会变成搜索计划程序（| archivebuckets）返回的事件。

完成或中止归档过程

当 Hadoop Data Roll 暂停或完成搜索时，此信息必须传递给下游的进程。

例如，如果索引器上的搜索进程关闭，搜索可能会终止子进程，这会阻止索引器的 Python 进程正常地关闭。如果 Python 进程使用的是共享资源，如数据库连接或到 HDFS 的输出流，这可能会导致故障及数据丢失。

要解决该问题，搜索进程允许子进程决定当搜索进程暂停或关闭时应该如何操作。如果索引器上的搜索进程暂停，则它会停止向索引器上的 Python 进程读取管道信息。当此情况发生时，一旦缓冲区填满，则索引器上的 Python 进程就无法向管道写入信息。

Python 进程可以确定索引器上的搜索进程仍然存在，但已暂停。

如果索引器上的搜索进程停止/完成，则它会关闭，并且连接到 Python 搜索进程的管道会断开。Splunk 自定义搜索命令以此方式知道上游搜索已停止运行。无论是由于用户操作而完全关闭搜索，或者由于上游崩溃而突然关闭搜索，都会出现这种情况。

如果索引器上的归档 Java 进程发现连接到索引器搜索进程的管道断开，则会记录该信息，但是会继续完成归档，直到缓冲区填满。如果这不是您想要的，只需要终止 Java 进程。

添加或编辑 Splunk Web 内的 HDFS 提供程序

您可以为一个提供程序设置多个含多个索引的提供程序。了解并掌握以下信息：

- Hadoop 群集的主机名和 NameNode 端口。
- Hadoop 群集的主机名和 JobTracker 端口。
- Hadoop 命令行库和 Java 安装的安装目录。
- DataNode/TaskTracker *nix 文件系统上可写目录的路径，Hadoop 用户帐号拥有该目录的读写权限。
- HDFS 中可写目录的路径，该目录在此搜索头上仅供 Splunk 使用。

您也可以通过编辑 indexes.conf 添加 HDFS 提供程序。

添加提供程序

1. 在顶部菜单中选择**设置** > **虚拟索引**。

2. 选择**提供程序**选项卡并单击**新提供程序**或您想要编辑的提供程序名称。

3. “添加新/编辑提供程序”页面可让您为提供程序**命名**。

4. 在下拉列表中选择**提供程序系列**（请注意，此字段无法编辑）。

5. 提供下列**环境变量**：

- **Java Home**：提供 Java 实例的路径。
- **Hadoop Home**：提供 Hadoop 客户端目录的路径。

6. 提供以下 **Hadoop 群集信息**：

- Hadoop 版本：指定群集运行的 Hadoop 版本：Hadoop 1.0、带 Mrv1 的 Hadoop 2.0 或带 Yarn 的 Hadoop 2.0。
- **JobTracker**：提供任务追踪器的路径。
- **文件系统**：提供默认文件系统的路径。

7. 提供以下设置：

- **HDFS 工作目录**：该路径位于 HDFS 或默认文件系统（无论它是什么）内，您想要把 HDFS 或默认文件系统用作工作目录。
- **任务队列**：您想把此提供程序的 MapReduce 任务提交到该任务队列中。

8. 单击**添加安全群集**为群集配置安全性，并提供您的 Kerberos 服务器配置。

9. **其他设置**字段指定您的提供程序配置变量。Hadoop Data Roll 为您创建的每个提供程序填充这些预设配置变量。您可以保留预设变量，或者根据需要编辑它们。如想了解更多有关这些设置的信息，请参阅本手册参考部分里的“提供程序配置变量”。

注意：若为了使用 YARN 而配置 Splunk Analytics for Hadoop，您必须添加新的设置。请参阅本手册中的“YARN 必需的配置变量”。

9. 单击**保存**。

使用配置文件把 Splunk 索引归档配置到 Hadoop

开始之前请先注意以下事项：

- 您必须配置一个 Hadoop 提供程序。
- 必需使用和所有索引器及 Splunk Enterprise 实例相同的用户安装 Splunk。该用户连接至 HDFS 进行归档，且用户和用户权限必需一致。
- 引用索引中的数据必需只能位于温、冷或冻结的数据桶内。
- Hadoop 客户端库必需和索引器位于相同的位置。同样地，Java 运行时间环境也必需安装在索引器上的相同位置。有关必需版本的更新信息，请参阅“系统和软件要求”。
- 与 Splunk 索引器相关的 Splunk 用户必需拥有把数据写入到 HDFS 节点的权限。
- 若数据桶中的原始数据大于 5GB，Splunk 目前还无法将其归档到 S3。您可以在 `indexes.conf` 中配置您的 Splunk Enterprise 数据桶大小。把数据归档至 S3 时的已知问题请参阅本手册中的“把 Splunk 索引归档至 S3”。

设置软件包删除参数

使用下面的属性来指定在 Splunk Enterprise 删除软件包之前，可以产生多少个软件包：

```
vix.splunk.setup.bundle.reap.limit = 5
```

默认值是 5，这意味着，当软件包超过五个时，Splunk Enterprise 将删除最早的一个。

在配置文件中配置索引归档

在 `indexes.conf` 中配置以下段落：

```
[splunk_index_archive]
vix.output.buckets.from.indexes
vix.output.buckets.older.than
vix.output.buckets.path
```

vix.provider

其中：

- vix.output.buckets.from.indexes 正是您想要复制到归档中的 Splunk 索引的名称。例如："splunk_index."您可以列出多个 Splunk 索引，中间用逗号隔开即可。
- vix.output.buckets.older.than 为 Splunk 索引内的数据桶数据归档前应存放的时间。例如，若您指定的时间为 432000 秒（5 天），数据将在存放五天之后被复制入归档。请注意，Splunk 会根据索引设置在一段时间之后删除数据，因此请确保数据在从 Splunk Enterprise 索引器中删除之前索引设置已复制了 Splunk 数据。
- vix.output.buckets.path 为 HDFS 中的目录，归档数据桶应存储在该目录中。例如："/user/root/archive/splunk_index_archive"。如果使用的是 S3，您应该把 s3n://<s3-bucket>/ 添加为该值的前缀，并从下面的代码示例中添加其他属性。
- vix.provider 为新归档的虚拟索引提供程序。

如果是 S3 目录，您必须为 vix.output.buckets.path 添加前缀，添加的内容为 s3n://<s3-bucket>/，然后再把以下列出的其他属性添加到提供程序段落：

```
vix.fs.s3n.awsAccessKeyId = <your aws access key ID>
vix.fs.s3n.awsSecretAccessKey = <your aws secret access key>
```

限制用于归档的带宽

您可以设置带宽限制，由此来限制归档的传输速率。

您为提供程序设置限制后，该限制将被应用到分配至该提供程序的所有归档。为配置限制，请把以下属性添加到您想要限制的虚拟索引提供程序段落下。

```
vix.output.buckets.max.network.bandwidth = <bandwidth in bits/second>
```

更多有关在 indexes.conf 中配置提供程序的信息，请参阅“在 Splunk Web 中添加或编辑提供程序”。

把 Splunk 索引归档至 Splunk Web 中的 Hadoop

开始之前请先注意以下事项：

- 您必须配置一个 Hadoop 提供程序。
- 必需使用和所有索引器及 Splunk Enterprise 实例相同的用户安装 Splunk。该用户连接至 HDFS 进行归档，且用户和用户权限必需一致。
- 引用索引中的数据必需只能位于温、冷或冻结的数据桶内。
- Hadoop 客户端库必需和索引器位于相同的位置。同样地，Java 运行时间环境也必需安装在索引器上的相同位置。有关必需版本的更新信息，请参阅“系统和软件要求”。
- 与 Splunk 索引器相关的 Splunk 用户必需拥有把数据写入到 HDFS 节点的权限。
- 若数据桶中的原始数据大于 5GB，Splunk 目前还无法将其归档到 S3。您可以在 indexes.conf 中配置您的 Splunk Enterprise 数据桶大小。把数据归档至 S3 时的已知问题请参阅本手册中的“把 Splunk 索引归档至 S3”。

使用用户界面配置索引归档

1. 导航至**设置** > **虚拟索引**并选择**已归档索引**选项卡。把箭头单击至其左侧可以编辑现有的任何已归档索引。

2. 单击**新归档索引**可以归档另一个索引。

3. 键入您想要归档的索引的名称。您可以添加多个索引。下拉列表中禁用了已归档的索引。

4. 为新的归档索引添加后缀。例如，若您选择了 "_archive" 后缀，新的归档索引将为 "indexname_archive"。

5. 选择新归档索引将被分配到其中的 Hadoop **提供程序**。

注意：您可以依照提供程序决定这些归档可以使用的带宽。请参阅本主题中的“设置归档的带宽限制”。

6. 把 **HDFS 中的目标路径**提供给您的提供程序要用于此数据的工作目录。例如：/user/root/archive/splunk_index_archive。若要把数据复制到 S3，请把以下内容添加为此路径的前缀： s3n://<s3-bucket>/

7. 确定数据要复制到归档索引前的存放时间。例如，若您选择了“5 天”，数据将在存放五天之后从索引器内的温、冷或冻结的数据桶复制到归档数据桶。**注意**：Splunk 将在您于索引器设置中定义的时间到期后删除数据，因此，请确保您将此字段设置

为在数据被删除前复制数据桶。

设置归档的带宽限制

若您担心持续归档所需的带宽，您可以设置带宽限制。您为提供程序设置限制后，该限制将被应用到分配至该提供程序的所有索引。

注意：我们当前无法保证 S3 文件系统的数据桶归档带宽限制。

带宽限制设置步骤如下：

1. 在“已归档索引”选项卡中为您想要编辑的索引单击**最大带宽（提供程序）**。此操作将打开该索引的“编辑提供程序”页面。

2. 在“归档设置”下勾选**启用归档带宽限制**。

3. 为与提供程序相关的所有已归档索引输入您想要设置的最大带宽。

4. 单击**保存**。

把 Splunk 索引归档至 S3 上的 Hadoop

为获得最佳性能并避免数据桶大小限制，您应使用 Apache Hadoop 2.6.0 中介绍的 S3A 文件系统。不同 Hadoop 分布的配置可能会相异。

把归档配置到 Amazon S3

若要将 Splunk 数据归档到 S3，您需要启用 Splunk Enterprise，具体步骤如下：

1. 将以下配置添加到您的提供程序：

```
vix.env.HADOOP_HOME: /absolute/path/to/apache/hadoop-2.6.0
vix.fs.s3a.access.key: <AWS access key>
vix.fs.s3a.secret.key: <AWS secret key>
vix.env.HADOOP_TOOLS: $HADOOP_HOME/share/hadoop/tools/lib
vix.splunk.jars: $HADOOP_TOOLS/hadoop-aws-2.6.0.jar,$HADOOP_TOOLS/aws-java-sdk-1.7.4.jar,$HADOOP_TOOLS/jackson-databind -
2.2.3.jar,$HADOOP_TOOLS/jackson-core-2.2.3.jar,$HADOOP_TOOLS/jackson-annotations-2.2.3.jar
```

2. 使用以上配置创建一个归档索引，其路径的前缀为 `s3a://`：

```
s3a://bucket/path/to/archive
```

在本示例中，

- `s3a` 为 Hadoop 从上述路径中传输和读取文件时使用的实现
- `bucket` 为您的 S3 数据桶的名称
- `/path/to/archive` 为数据桶内的目录

唯一设置的进一步配置

您可能需要进一步配置 Splunk Enterprise 才能根据您配置的详细信息搜索 S3 归档数据。

如果您仅使用一个搜索头

如果您使用一个搜索头来搜索归档数据，请将提供程序的 `vix.mode` 属性设置为 `stream`：

```
vix.mode = stream
```

把 `vix.mode` 设置为 `stream` 时，Splunk Enterprise 把搜索匹配到的所有数据以数据流的形式传入搜索头，且不会在 Hadoop 上衍生 MapReduce 任务。

如果您已使用 Hadoop 群集配置了搜索头

如果搜索头归档索引的 Hadoop 版本可与您的 Hadoop 群集兼容，则搜索您的归档索引时无需其他配置。您只需前往 Splunk Web 搜索栏并输入：

```
index=<your-archive-index-name>
```

搜索头将在合适的时候针对您的归档数据衍生 Hadoop MapReduce 任务。

如果您的 Hadoop 群集版本与您的 Hadoop Home 版本不兼容

就算您的 Hadoop 群集与 Hadoop 客户端库（带 S3a 文件系统）不兼容，您仍可以使用 Data Roll。此情况的一个示例为：您使用 Apache Hadoop 2.6.0 为数据归档但将 Hadoop 1.2.0 用于 Hadoop 群集。如需在上述情况下使用 Data Roll，请使用较旧的 S3n 文件系统来搜索您的归档。

如需搜索您的归档，请按下列步骤配置 S3n 和较旧的 Hadoop 群集：

1. 为您的 Hadoop 群集配置一个提供程序。

2. 从您的终端中为每个归档索引配置 indexes.conf 并添加具有以下属性的新虚拟索引：

```
[<virtual_index_name_of_your_choosing>]
vix.output.buckets.path = <archive_index_destination_path_with_s3n_instead_of_s3a>
vix.provider = <hadoop_cluster_provider>
```

3. 确保 vix.output.buckets.path 为 S3n，这样 Splunk Enterprise 才能使用较旧的文件系统来搜索您的归档。

例如。假设一个归档索引名为 "main_archive"，目标路径为 "s3a://my-bucket/archive_root/main_archive" 且提供程序 = "hadoop_cluster"，您应该按以下方式配置虚拟索引：

```
[main_archive_search] vix.output.buckets.path = s3n://my-bucket/archive_root/main_archive vix.provider = hadoop_cluster
```

S3 的已知问题

使用 Hadoop 的 S3N 文件系统时，您只能上载大小不超过 5GB 的文件。虽然 Splunk 数据桶可能会超过 5GB，但这种情况很少发生。使用 S3N 文件系统时，超过 5GB 的数据桶将无法归档。

如果使用的是 S3N 文件系统，请配置您的索引，以通过 maxDataSize 属性（位于 indexes.conf 中）把小于 5GB 的数据桶从热滚动到温。

Data Roll 归档的最低要求为写入后读取一致。对于采用美国标准的地区而言，仅当通过北弗吉尼亚端点访问时才能保证写入后读取一致。更多详细信息请参阅“Amazon AWS S3 常见问题”。

更多有关使用 S3a 进行归档的信息，请访问 此博客介绍如何使用 S3A 快速、无限制地归档。

数据桶原始数据限制

由于 Hadoop 与 S3 文件系统之间的交互方式，Splunk Enterprise 目前无法把原始数据集大于 5GB 的数据桶归档到 S3。

我们建议使用支持 5GB 以上文件上载的 S3FileSystem 实现。为确保所有数据都可以归档，请配置您的索引，以通过 maxDataSize 属性（位于 indexes.conf 中）把小于 5GB 的数据桶从热滚动到温。

数据复制进程

数据被归档到 S3 时会复制两次。这是因为 S3 不支持文件重命名，而 FileSystem 则按以下方式实现文件重命名：

- 下载文件
- 重命名后上载文件
- 删除原始文件

此进程不会在您的归档中创建重复的数据。

带宽限制的局限性

Splunk Enterprise 无法确保把数据归档到 S3 时会遵守带宽限制。如果有配置，Splunk 仍会在可能的情况下尝试限制带宽。

搜索归档到 Hadoop 的索引数据

适当安装并配置归档索引后，您就可以新建报表，并像在传统的 Splunk 索引中一样对数据进行可视化处理。使用虚拟索引和传统的 Splunk Enterprise 之后，您可以只从虚拟索引中收集数据，或者也可以同时查询本地索引和虚拟索引并制作单份报表。

大多数情况下，您既可以新建虚拟索引报表也可以新建本地索引报表。更多有关新建报表的信息，请参阅*《Splunk Enterprise 搜索手册》*。

由于事件并未排序，任何基于隐式时间顺序的搜索命令都无法达到您预期的效果。（例如：头、增量或交易。）这意味着有些搜索命令在用于虚拟索引时会以不同的方式运行，主要取决于 Hadoop 报告时间戳的方式。

您仍可以使用这些命令，尤其是需要为本地和虚拟索引新建单份报表时，但请注意这些索引如何以不同的方式运行并以不同的方式返回数据。

搜索语言

大多数情况下，您可以使用 Splunk Enterprise 的搜索语言新建报表。但是，由于 Hadoop 不支持事件顺序的严格要求，因此会产生一些差异。

当搜索包含归档索引时，将不支持下列命令：

- transactions
- localize

下列命令可以在已归档的索引上使用，但结果可能会不同于 Splunk。这是因为 Hadoop 不保证事件按时间降序排列。

- streamstats
- head
- delta
- tail
- reverse
- eventstats
- dedup （由于命令无法在 HDFS 目录内区分挑选移除项目的顺序，Splunk Analytics for Hadoop 将根据修改时间或文件顺序挑选要移除的项目。）

归档中的可分布式和不可分布式命令

可分布式搜索命令是 Hadoop Data Roll 回报的命令中最有效的，因为它们可以被分布到搜索对等节点和归档索引上。一般而言，不可分布式命令仅用于本地索引，在已归档索引上的效果不如前者有效。

您可以在同时使用可分布式和不可分布式命令的不同索引类型上新建各种搜索，但您需要记住，这样的搜索会返回本地索引上的所有数据，但仅返回虚拟索引上的有限数据。

使用虚拟索引时要避免的标头提取

已归档的索引不支持索引时间字段的配置。因此，索引时间字段提取特有的属性不适用于归档索引。具体包括下列属性：

- INDEXED_EXTRACTIONS
- HEADER_FIELD_LINE_NUMBER
- PREAMBLE_REGEX
- FIELD_HEADER_REGEX
- FIELD_DELIMITER
- FIELD_QUOTE
- HEADER_FIELD_DELIMITER
- HEADER_FIELD_QUOTE
- TIMESTAMP_FIELDS = field1,field2,...,fieldn
- FIELD_NAMES
- MISSING_VALUE_REGEX

把 Hadoop 中的冷数据桶归档为冻结数据桶

每个索引器上的数据都会在本地老化。您配置索引的方式决定着数据的大小或数据在移动到下一个状态（热、温、冷、冻结）并最终被删除之前的存放时间。

一旦您为数据归档配置好一个索引后，各索引的归档将按计划运行，该计划是在 Splunk 搜索头上全局决定的。

当本地进程和归档进程同时发生时，索引器的这两个进程之间会断开连接。因此，各索引器可以在数据桶归档之前将其删除。

为避免删除数据桶，您可以使用本地索引器进程上的 splunk_archiver 应用的 coldToFrozen.sh 脚本。此脚本会把删除数据桶的责任从索引器转移到 Hadoop Data Roll，因此仅对已归档的索引使用此脚本。

将 coldToFrozen.sh 脚本视为回退而非归档的主要框架。无论您的系统接收数据的速度比常规较快，还是归档存储层已关闭，此脚本都可以为您赢得更多的时间，让您可以有更多的时间来归档指定的数据桶。为进一步协助归档进程，您可以针对每个归档

索引把 `vix.output.buckets.older.than = seconds` 设置得尽可能低，这样就可以尽可能地加快数据桶归档的速度。

配置冷数据桶以将其滚动到冻结

若您使用的是 `coldToFrozen.sh` 脚本，请注意以下事项：

- 脚本必须安装在每一个段落上，这些段落配置正在进行归档的索引。
- 搜索头的所有搜索对等节点必须安装脚本。您可以手动为每个对等节点安装脚本，也可以针对搜索头群集使用 Deployer。
- 必须从您已禁用归档的索引中删除脚本。否则，脚本将继续运行，导致数据塞满您现有的磁盘空间，因为没有归档会接收数据（数据因此不会被删除）。
- 切勿将此脚本添加到任何未进行数据归档配置的索引器。

就每个 Splunk 索引而言，如需把您的冷数据归档为冻结数据，请使用提供的脚本，该脚本位于 `$SPLUNK_HOME/etc/apps/splunk_archiver/bin/` 中且被命名为 `coldToFrozen.sh`。此路径可能会因为您的配置路径而有所不同。例如：

```
[<index name>]
coldToFrozenScript = "$SPLUNK_HOME/etc/apps/splunk_archiver/bin/coldToFrozen.sh"
```

Hadoop Data Roll 故障排除

问题：Splunk 用户权限问题干扰了将数据从索引器拷贝到 HDFS 的能力

数据归档在索引器上进行。但是，所有的归档设置都是在搜索头上完成的。搜索头和索引器之间的权限问题会阻止归档。

例如，如果索引器权限与搜索头不一致，您可能会看到该异常：

```
java.io.IOException: Login failure for null from keytab /etc/security/keytabs/splun.keytab:
javax.security.auth.login.LoginException: Unable to obtain password from user
```

您可以作为 Splunk 用户，通过从索引器拷贝文件到 HDFS 来测试权限： `hadoop fs -put somefile /user/splunk/archive`

检查以下方面以保证 Splunk 用户始终配置有 HDFS 权限：

- Kerberos Keytab 路径
- Hadoop 客户端库路径
- Java 库路径
- Splunk 用户在索引器上存在。
- Splunk 用户具备向 HDFS 写入的权限。

问题：您需要收集归档错误

以通用的方式捕获所有的归档错误，然后发出告警。

要打开归档仪表板进行调试，请访问：*设置 > 虚拟索引 > 归档的索引 > 查看仪表板*

Splunk Web 使用以下查询：

```
index=_internal source=*splunk_archiver.log* earliest=-1d | rex max_match=1000 "\d{4}-\d{2}-\d{2} \d{2}:\d{2}:\d{2}\.\d+ -\d{4}
(?<severity>\w+) " | where severity="ERROR"
```