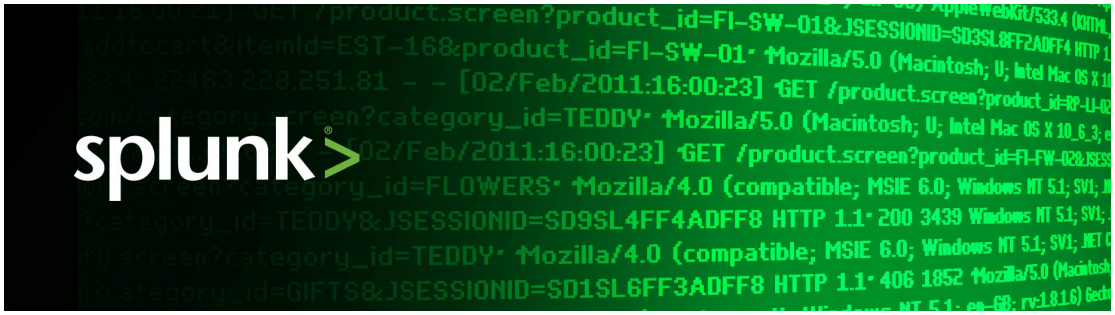


Splunk® Enterprise 6.5.0

管理索引器和索引器群集

生成时间：2016 年 9 月 26 日，下午 10:25

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

Splunk Enterprise 为您的数据创建索引时，会创建许多文件。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。

创建索引期间，Splunk Enterprise 对传入数据进行处理，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、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、过滤出不需要的事件以及将事件路由到指定索引或服务器。

此索引过程也称为**事件处理**。

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。*《数据导入手册》*还介绍了如何配置索引时间事件处理，以满足您的数据需求。请参阅“事件处理概述”。

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。

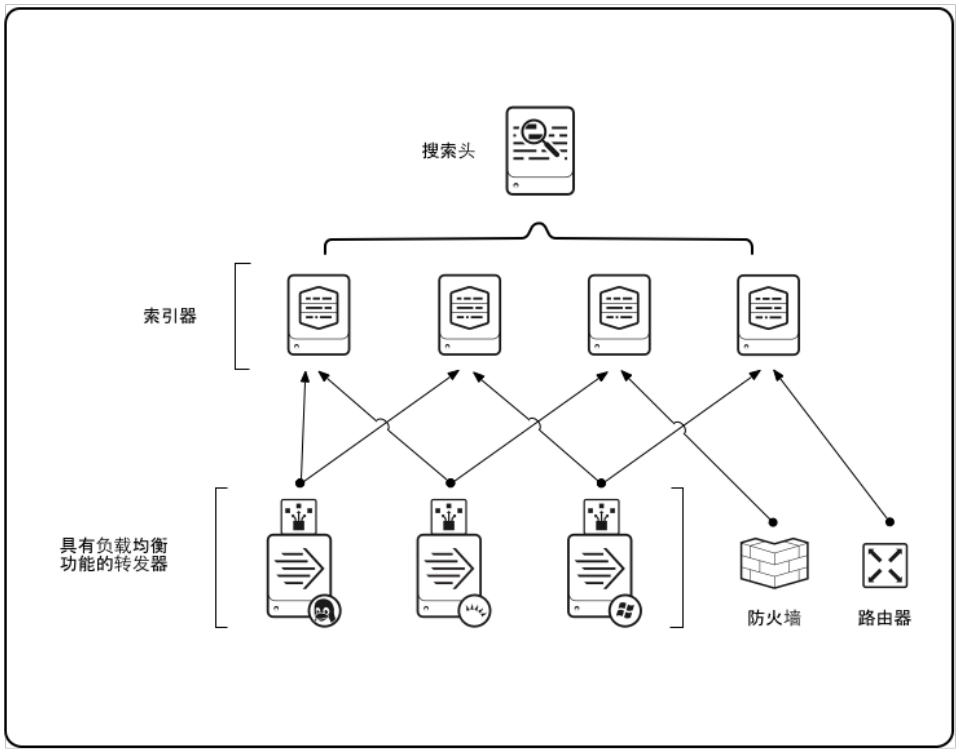
索引器

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**主节点**，用于管理群集。主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可继续对已索引数据执行搜索。

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序列数据（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**。

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：

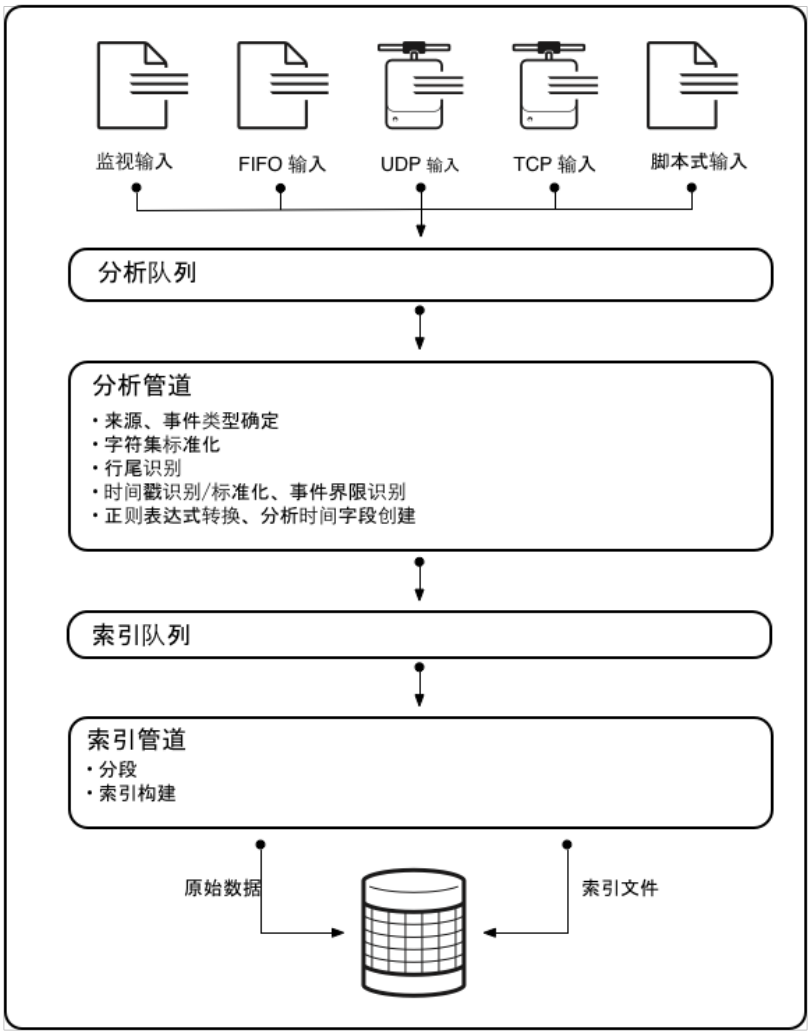
- 将所有事件分段，然后基于段执行搜索。您可以确定分段的级别，它将影响索引和搜索速度、搜索功能以及磁盘压缩效率。
- 构建索引数据结构。
- 将原始数据和索引文件写入磁盘，其中将执行后索引压缩。

分析与索引管道间的中断主要与部署的**转发器**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创建。**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。但是，请注意，这两种类型的转发器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。

注意：索引创建是 I/O 密集型进程。

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 将其处理的所有数据存储在索引中。索引是位于 `$SPLUNK_HOME/var/lib/splunk` 中子目录的数据库的集合。索引由两种类型的文件组成：**原始数据文件**和**索引文件**。请参阅 Splunk Enterprise 如何存储索引。

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、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来管理问题。

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。

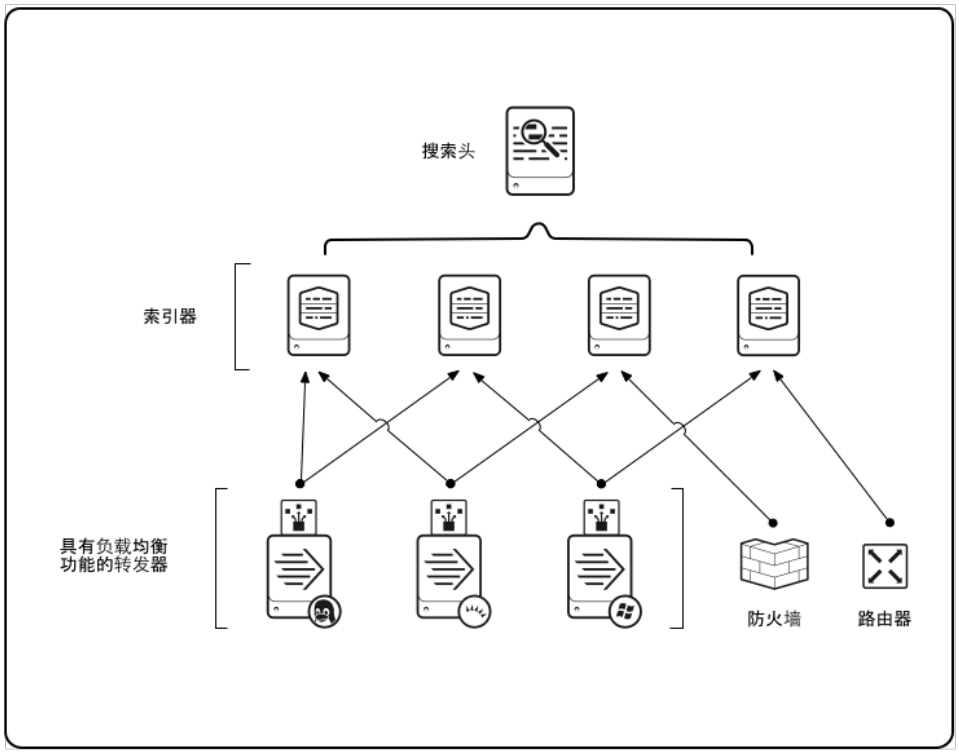
在仅包含一个 Splunk Enterprise 实例的单一计算机部署中，索引器还将处理**数据导入**和**搜索管理**功能。

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、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**筛选**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**简化更新部署组件的工作。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**：本索引包括 Splunk Enterprise 内部日志和指标。
- **_audit**：来自文件系统更改监视、审计和所有用户搜索历史的事件。

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

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

默认情况下，`main` 索引保存您的所有事件。索引器还拥有许多其他索引，供其内部系统使用以及用于其他功能（如摘要索引和事件审计）。

具有 Splunk Enterprise 许可证，您就能添加无限多个其他索引。`main` 索引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。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/高噪音数据源和小容量数据源提供给同一个索引，并且您主要从小容量数据源中搜索事件，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。

创建和编辑索引

可以通过 Splunk Web、CLI 或直接编辑 `indexes.conf` 来创建或编辑索引。

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

1.在 Splunk Web 中，导航到**设置 > 索引**，然后单击**新增**。

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 `$SPLUNK_DB/<index_name>/db`
  - 冷数据库路径；保留空白，以使用默认值 `$SPLUNK_DB/<index_name>/colddb`
  - 解冻/恢复数据库路径；保留空白，以使用默认值 `$SPLUNK_DB/<index_name>/thaweddb`
- 整个索引的最大大小。默认为 500,000MB。
- 此索引的热部（当前写入到的部分）的最大大小。设置最大大小时，应对大容量索引（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。
- tsidx 保存策略。请参阅减少 tsidx 使用。

注意： 有关上述每项设置的详细信息，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。

重要提示： 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>` 必须为您在步骤 1 中确定的属性提供一个匹配。
- 必须将 `DEST_KEY` 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。

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。

警告：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特殊运算符 `delete`。在使用 `delete` 之前，请仔细阅读本部分内容。

注意：您不能在实时搜索期间运行 `delete` 运算符；您不能在事件数据传入时将事件删除。如果您试图在实时搜索期间使用 `delete`，Splunk Enterprise 将显示一条错误信息。

什么用户可以执行删除操作？

只有具有 "delete_by_keyword" **操作**的用户可以运行 `delete` 运算符。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，可以将此搜索的结果通过管道符传递给 `delete` 运算符。

例如，如果想要删除根据名为 `/fflanda/incoming/cheese.log` 的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.禁用或删除该来源，以便不再为其建立索引。

2.在您的索引中搜索来自该来源的事件：

```
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 运算符相关的页面。

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` 运算符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` 运算符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**中永久删除事件数据，键入：

```
splunk clean eventdata
```

- 要从**单个索引**中永久删除事件数据，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事件数据：

```
splunk stop
splunk clean eventdata
```

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

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

删除整个索引

要删除整个索引（而不仅仅是其中包含的数据），使用 CLI 命令 `remove index`：

```
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。

运行此命令之前，请先查看所有 `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。换句话说，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。

在运行 `remove index` 时，它会首先向您发出警告消息，要您确认索引器上（而不是在任何转发器上）是否存在仍配置为将数据发送到指定索引的输入。您将看到如下消息：

```
03-28-2012 23:59:22.973 -0700 WARN  IndexAdminHandler - Events from the following 3 inputs will now be discarded,
since they had targeted index=zzz:
03-28-2012 23:59:22.973 -0700 WARN  IndexAdminHandler - type: monitor, id: /home/v/syslog-avg-1000-lines
03-28-2012 23:59:22.973 -0700 WARN  IndexAdminHandler - type: monitor, id: /mnt/kickstart/internal/fermi
03-28-2012 23:59:22.973 -0700 WARN  IndexAdminHandler - type: monitor, id: /mnt/kickstart/internal/flights
```

可以将 `remove index` 与 `splunkd` 同时运行。在命令完成时，不需要重新启动 `splunkd`。

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。

禁用索引（而不删除）

使用 `disable index` CLI 命令来禁用索引，而不将其删除：

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

与 `remove index` 命令不同，`disable index` 不删除索引数据，该命令是可撤消的（通过 `enable index` 命令）。但是，禁用某个索引后，`splunkd` 将不再接受以此为目标的数据。

您也可以在 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**。

基于退休策略删除旧数据

索引中的数据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。删除数据之前，索引器可以将其移动到归档中，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。

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”。

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。

每个数据导入进入单个管道。例如，如果您直接获取一个文件，则整个文件都将通过单个管道进行处理。管道间不共享文件的数据。

当数据导入进入索引器的时候，它可以进入两个管道集中的任意一个。索引器使用循环负载均衡在其管道集间分配新的导入。

每个管道写入它自己的一组热数据桶中。

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。

转发器使用循环负载均衡在其管道集间分配新的导入。

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。您可以减少 `splunk-optimize` 的并发实例数，具体做法为更改 `maxConcurrentOptimizes` 的值（位于 `indexes.conf` 中），但通常没有必要这样做。

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
```

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，您还可以参阅*《监视 Splunk Enterprise 手册》*中的“索引性能：实例”和“索引性能：部署”。

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。这些文件包含两种类型的数据：

- 压缩形式的原始数据（**原始数据**）
- 指向原始数据的索引加上部分元数据文件（**索引文件**，也称为 **tsidx 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中。部分目录包含新建索引的数据；其他目录包含以前索引的数据。此类目录的数量会变得相当大，取决于您要创建索引的数据量。

可能关心的原因

实际上您可能不关心。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，则必须阅读本主题。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

每个索引目录称为**数据桶**。到目前为止的汇总结果：

- “索引”包含压缩的原始数据和相关的索引文件。
- 索引驻留在时间指定的许多索引目录中。
- 一个索引目录称为一个数据桶。

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。当数据被索引时，它就进入一个热数据桶。可以对热数据桶执行搜索以及主动向其中写入内容。一个索引每次可以打开若干个热数据桶。

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 `splunkd`）时，热数据桶将变为温数据桶（“滚动到温数据桶”），并将在同一位置创建一个新的热数据桶。温数据桶是可搜索的，但不能主动向其中写入内容。会有许多个温数据桶。

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。在经过设定的一段时间后，冷数据桶滚动到冻结数据桶，此时对它们执行存档或删除操作。通过编辑 indexes.conf 中的属性，可指定数据桶老化策略，用于确定数据桶何时从一个阶段移动到下一个阶段。

如果冻结的数据已归档，则稍后它可以被解冻。解冻的数据可用于搜索。

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	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。数据不会主动写入温数据桶。	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，但您可改为选择将其归档。可在以后将已归档数据解冻。	否
解冻	数据从存档中恢复。如果您将冻结的数据归档，则稍后可以通过将其解冻使其返回到索引中。	是

有时，将某特定阶段中的数据桶集合称为数据库或 "db" ：“热 db”、“温 db”、“冷 db”等等。

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。在索引目录下是一系列子目录通过阶段（热/温、冷或解冻）分类数据桶。

数据桶本身就是这些目录的子目录。数据桶目录名称基于数据存在的时间。

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/*	可存在多个热子目录，每个对应一个热数据桶。请参阅"数据桶命名约定"。
温	$SPLUNK_HOME/var/lib/splunk/defaultdb/db/*	每个温数据桶都有单独的子目录。请参阅"数据桶命名约定"。
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息，请参阅"恢复归档的数据"。

热/温、冷和解冻目录的路径是可配置的，因此，例如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。

注意：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。

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，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化的介绍，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 日上午 9 点到上午 10 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"。

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 indexes.conf 的副本（位于 $SPLUNK_HOME/etc/system/local/ 或 $SPLUNK_HOME/etc/apps/ 中的自定义应用目录）。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"。
  也可以使用 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。此方法仅适用于非群集索引。
- **对于群集索引，**编辑群集主节点上的 indexes.conf 副本，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所述。

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。

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）	对索引数据使用多个分区
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）	对索引数据使用多个分区
	包含所有解冻数据桶的路		

thawedPath	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）	对索引数据使用多个分区
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	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。	视情况而定；请参阅 indexes.conf。	设置退休和归档策略
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

索引大小和索引器群集

控制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主节点。从此时起，主节点就不会再为冻结的数据桶发起任何修复活动。但是，主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。

移动索引数据库

可将整个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。

本主题中的过程会假定索引数据库位于其默认位置，而且是在原始安装期间创建的。

也可将各个索引或索引的各个部分移动到不同的位置。执行此操作后，本主题中的过程将不再有效。有关 Splunk Enterprise 索引结构的详细信息，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

1.确保目标文件系统有足够的空间 - 至少是您计划创建索引的原始数据总大小的 1.2 倍。

2.创建目标目录，确保 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 运行时所使用的用户的信息，请参阅本主题。

3.如果新索引主目录已准备好，请停止索引器。导航到 `$SPLUNK_HOME/bin/` 目录，然后运行以下命令：

```
splunk stop
```

4.将现有索引文件系统复制到新的主目录：

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

5.取消设置 `SPLUNK_DB` 环境变量：

```
unset SPLUNK_DB
```

6.编辑 `$SPLUNK_HOME/etc/splunk-launch.conf` 以反映该新索引目录。将该文件中的 `SPLUNK_DB` 属性更改为指向您的新索引目录：

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

7.启动索引器。导航到 `$SPLUNK_HOME/bin/`，然后运行以下命令：

```
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。

8.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

1.确保目标驱动器或目录具有足够的可用空间。

警告：我们不建议且不支持将映射的网络驱动器用于索引存储。

2.在命令提示符下，转到您的目标驱动器并确保目标目录具有正确的权限，这样，`splunkd` 进程能够写入文件，其中：

```
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 运行时所使用的用户的详细信息，请参阅本主题关于在 Windows 上安装 Splunk 的部分。

3.停止索引器。导航到 `%SPLUNK_HOME%\bin` 目录，然后运行以下命令：

```
splunk stop
```

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

4.将现有索引文件系统复制到新的主目录：

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

5.取消设置 `SPLUNK_DB` 环境变量：

```
set SPLUNK_DB=
```

6.编辑 `%SPLUNK_HOME%\etc\splunk-launch.conf` 以反映该新索引目录。将该文件中的 `SPLUNK_DB` 属性更改为指向您的新索引目录：

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

注意：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，则该行已被注释掉，并需要删除 #。

7.启动索引器。导航到 `%SPLUNK_HOME%\bin` 目录，然后运行以下命令：

```
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。

8.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

索引器可对其索引数据使用多个磁盘和分区。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。不过，为获得最佳体验，我们建议您使用单个高性能文件系统来存储索引器索引数据。

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（进行长期存储）。您需要在具有快读取/写入分区的计算机上运行热/温数据桶，因为，大多数搜索将在其中进行。冷数据桶应位于可靠的磁盘阵列上。

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，也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。下面是可设置的路径属性：

- `homePath = <path on server>`
  - 此路径包含索引的热数据库和温数据库。
  - **警告：**该路径必须是可写路径。
- `coldPath = <path on server>`
  - 此路径包含索引的冷数据库。
  - **警告：**该路径必须是可写路径。
- `thawedPath = <path on server>`
  - 此路径包含索引的所有解冻数据库。

配置最大索引大小

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` 属性：

```
# 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。

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。有关 `bloomHomePath` 的信息，请参阅本手册中的“配置布隆过滤器”主题。

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 bucket 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

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

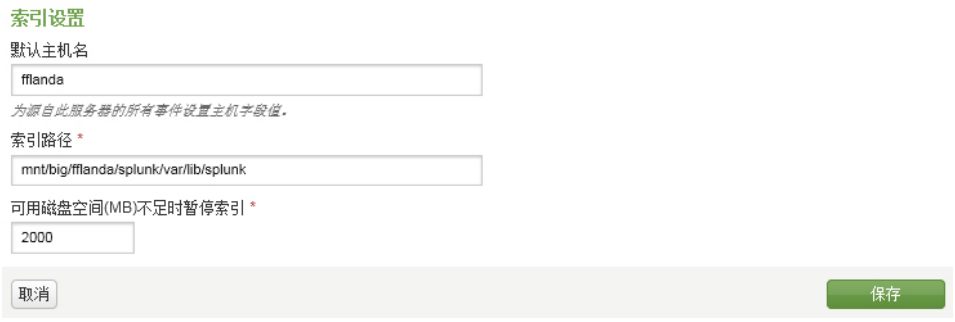
在 Splunk Web 中指定最小磁盘使用：

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，设置字段**可用磁盘空间 (MB) 不足时暂停索引**：



5.输入所需的最小可用磁盘空间 (MB)。

6.单击**保存**。

7.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

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 减少进程将删除最大大小的 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，可以将大小减少后的数据桶还原成带完整 tsidx 文件的数据桶。请参阅将大小减少后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。

注意：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。

注意：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后的数据桶后，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后的数据桶时，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。对缩小后数据桶的搜索速度可能会变慢。"

有几个搜索命令不适用于大小减少后的数据桶。这些命令包括 tstats 和 typeahead。当这种搜索触及大小减少后的数据桶时，search.log 中新增一个警告："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。"

注意：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

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>
```

请参阅“重新构建单个数据桶”。

4. 重新启动索引器。

配置布隆过滤器

本主题介绍布隆过滤器以及 Splunk Enterprise 如何使用布隆过滤器来提高搜索性能，尤其是改进罕见术语搜索的性能。

继续阅读本主题之前，您应了解索引器如何存储数据以及数据如何在索引后老化。基本上，已创建索引的数据驻留在包含称为**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（热、温、冷和冻结）。参阅有关“索引器如何存储数据”和“数据如何老化”的详细信息。

为何使用布隆过滤器？

布隆过滤器是一种数据结构，用于测试某个元素是否为某个集合的成员。我们的实现是将布隆过滤器作为一个文件存

储在磁盘上的每个数据桶中。运行搜索时，特别是在搜索罕见术语时，使用布隆过滤器会大大减少从索引中检索事件所需的时间。

在索引器为您的时间序列数据创建索引时，将创建一个压缩文件，其中包含基于时间戳拆分为多个事件的原始数据和一组**时间序列索引 (tsidx) 文件**。tsidx 文件是字典文件，它们相当于数据中所有关键字（错误代码、响应时间等）的字典，并包含到原始数据中事件位置的引用。运行搜索时，索引器将在 tsidx 文件中搜索关键字，并从引用的原始数据文件中检索事件。

布隆过滤器在数据桶级别工作并使用单独的文件 `bloomfilter`，该文件基本上是一个哈希表，能够告诉您某个关键字确实不在数据桶中。这样，当运行搜索时，索引器只需在布隆过滤器未排除的数据桶中搜索 tsidx 文件。从磁盘检索事件的执行成本会随着 tsidx 文件大小和数量的变化而发生变化。由于布隆过滤器会减少索引器需要搜索的 tsidx 文件的数量，因此会缩短搜索每个数据桶所花费的时间。

布隆过滤器不是将找到的所有唯一关键字都存储在数据桶的 tsidx 文件中，而是计算每个关键字的哈希值。多个关键字可能会产生同一个哈希值，这意味着可能会产生误报现象，但永远也不会存在漏报现象。因此，布隆过滤器能够快速排除确定没有存在于特定数据桶的术语，而且索引器将继续搜索下一个数据桶。如果布隆过滤器无法排除某个数据桶（关键字可能实际存在于该数据桶中，也可能不存在），索引器将正常搜索该数据桶。

配置布隆过滤器

当数据桶从热滚动到温时，将创建布隆过滤器。默认情况下，当数据桶滚动到冻结时，将删除这些过滤器，除非您配置了不同的保留行为。本部分介绍可用于配置和管理布隆过滤器文件的配置文件参数。

要指定是否需要使用布隆过滤器，请使用 `use_bloomfilter` 参数（位于 `limits.conf` 中）：

```
[search]
use_bloomfilter = true|false
* Control whether to use bloom filters to rule out buckets.
* Defaults to True.
```

要为某个特定索引创建布隆过滤器，请在 `indexes.conf` 中编辑以下**按索引**选项：

```
bloomHomePath = <path on indexer>
    * The location where the bloom filter files for the index are stored.
    * If specified, it must be defined in terms of a volume definition.
    * If not specified, bloom filter files for the index will be stored inside the bucket directories.
    * The path must be writable.
    * You must restart splunkd after changing this parameter.

createBloomfilter = true|false
* Determines whether to create bloom filter files for the index.
* Defaults to "true".
```

此外，如果显式定义了 `bloomHomePath`，则必须使用卷。一个卷代表文件系统上一个驻留已创建索引数据的目录。有关更多信息，请阅读"配置索引大小"。

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：

- 更改如下属性：`homePath`, `coldPath`, `thawedPath`, `bloomHomePath`, `summaryHomePath`, `tstatsHomePath`, `repFactor`, `rawChunkSizeBytes`, `minRawFileSyncSecs`, `syncMeta`, `maxConcurrentOptimizes`, `coldToFrozenDir`
- 添加或删除卷
- 启用带数据的索引
- 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
- 启用或禁用不带数据的索引

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

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"中的描述命名。
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和索引复制的环境中，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

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：设置 maxTotalDataSizeMB

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。

当数据时间太久时将其冻结：设置 frozenTimePeriodInSecs

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"中的描述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

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

索引器将根据设置退休和归档策略中所述的数据退休策略，把旧数据从索引中旋转出来。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。

群集数据归档

索引器群集包含索引数据的冗余副本。如果您使用上文所述的方法归档该数据，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，这在管理上有点困难，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相同。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。

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_1
/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，就可以再次对新解冻的数据桶中的内容进行搜索了。

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
/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，就可以再次对新解冻的数据桶中的内容进行搜索了。

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。

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，可继续为传入数据创建索引且可继续对索引数据执行搜索。

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。

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**。

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**主节点**，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。

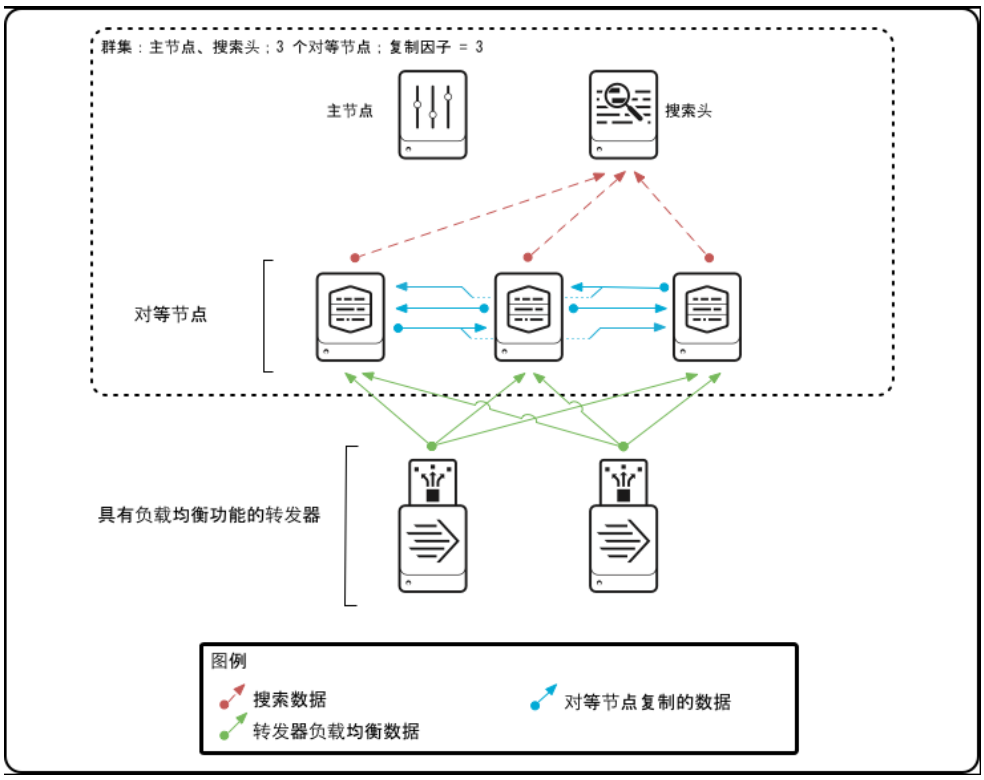
主节点会管理群集。它会协调对等节点之间的复制活动，并向搜索头指出查找数据的位置。它还会帮助管理对等节点的配置并在对等节点故障时安排补救活动。

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。
- 主节点，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 实例指定为主节点，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。

请参阅章节"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，搜索头将咨询主节点来确定当前的一组对等节点。然后，索引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执行搜索相关性。"

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：处理主站点故障，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。

群集节点

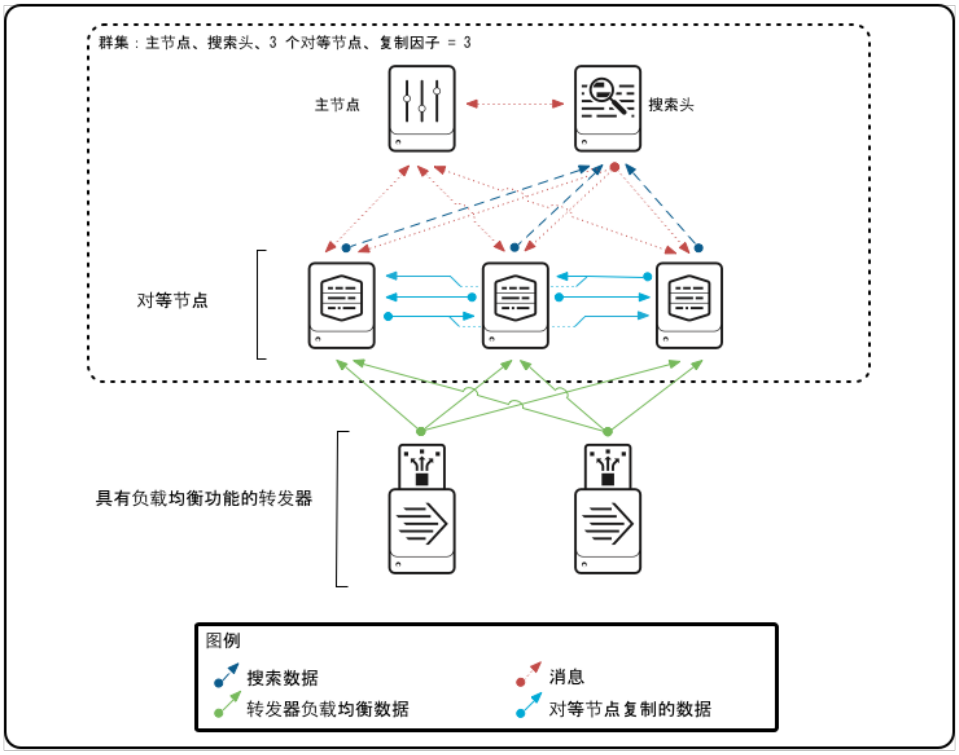
群集包括三种类型的节点：

- 单个**主节点**，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。

主节点、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，主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。

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。

主节点

主节点会管理群集。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，主节点不为**外部数据**建立索引。一个群集只有一个主节点。

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：

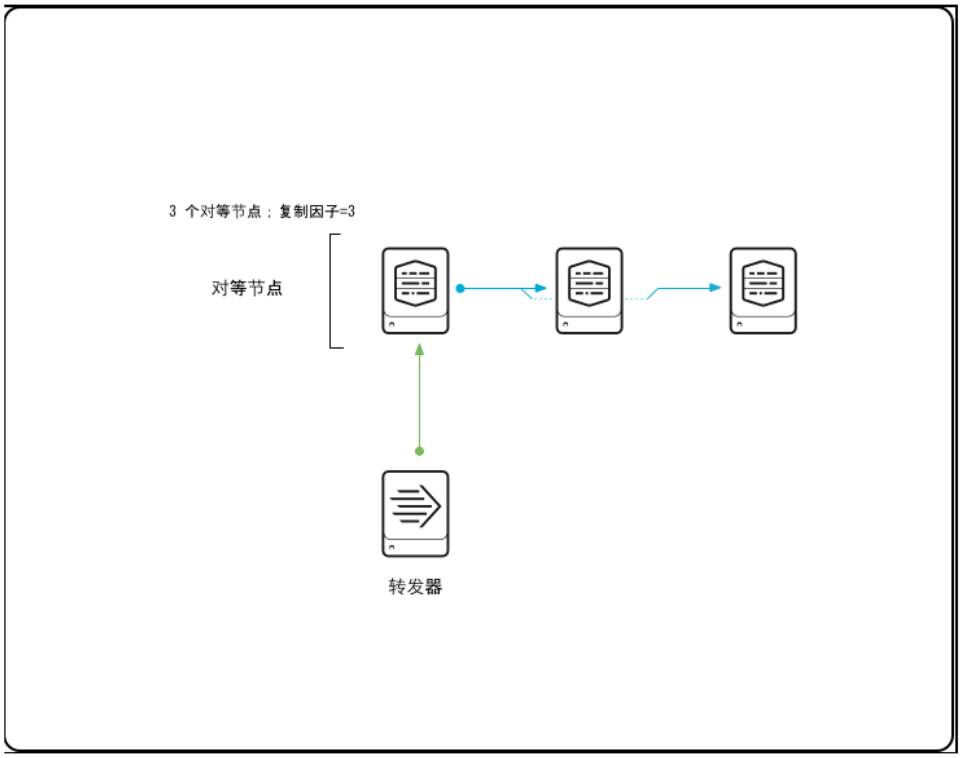
- **复制因子**。用于确定群集保留的数据副本数，因此，也确定了群集的故障容错基本级别。
- **搜索因子**。用于确定群集保留的可搜索的数据副本数，因此，也确定了在对等节点故障后群集恢复其搜索功能的速度。
- **数据桶**。数据桶是索引存储的基本单位。群集会保留与每个数据桶复制因子数相同的副本数。

本部分将简要介绍这些概念。

复制因子

在配置主节点时，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

配置主节点时，还应指定**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。

完整的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。完整的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。

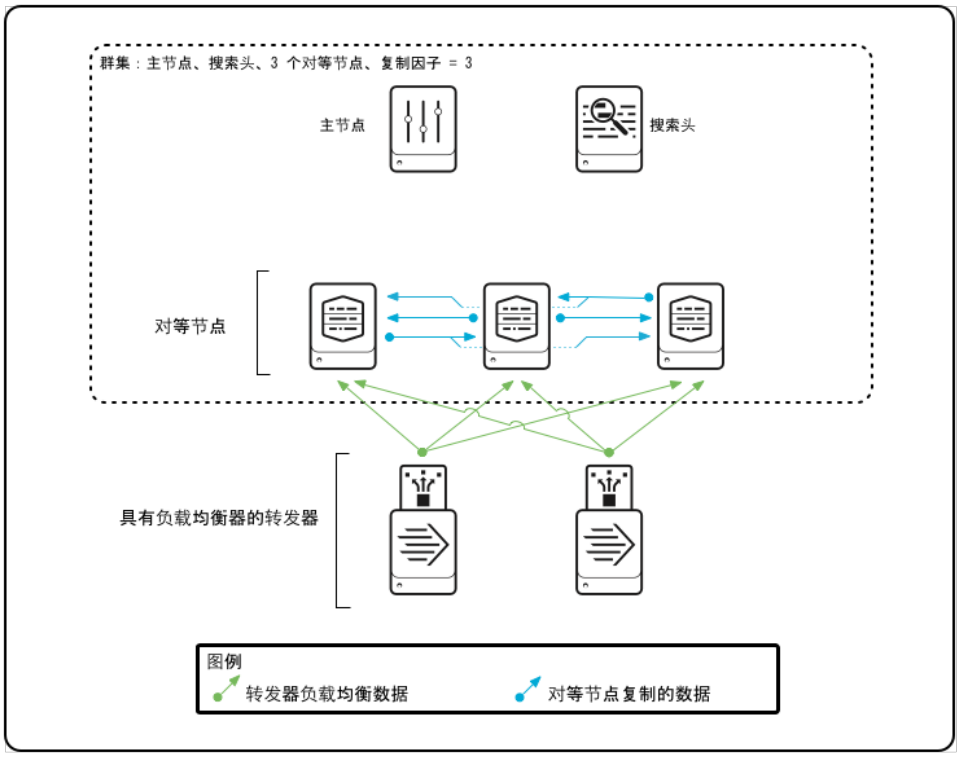
每当某个对等节点创建新数据桶时，它会与主节点通信，获取可将数据桶数据流入的对等节点列表。如果您的群集中对等节点的数量大于复制因子，则每当某个对等节点创建新数据桶时，它可能会让其数据流入另外一组对等节点。最终，该对等节点原始数据桶的副本可能分布在很多个对等节点上，即使复制因子仅为 3。

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。主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。

主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。一旦主节点完成通信任务，对等节点便彼此交换数据而无需主节点的涉入，除非某个对等节点出现故障。主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，主节点将协调补救活动。

有关详细信息，请阅读“群集式索引如何工作”主题。

请阅读“多站点索引”，获得关于索引在多站点群集中的工作方式的信息。

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，搜索头依赖主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，主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”。

主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。

搜索头会定期从主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。请阅读“多站点搜索”，获得更多信息。

群集如何处理对等节点故障

如果对等节点出现故障，主节点将进行协调，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，如果故障节点存储数据桶的 20 个副本，其中 10 个是可搜索副本（包括三个主要数据桶副本），主节点会直接在其他节点上创建这 20 个数据桶的副本。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，主节点将无法更换这 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。这允许主节点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主节点。

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，请阅读“对等节点故障时的情况”。

请阅读“多站点索引器群集如何处理对等节点故障”，获得关于多站点群集处理对等节点故障的信息。

群集如何处理主节点故障

如果主节点发生故障，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。没有主节点，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，没有主节点，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。

有关主节点故障的详细信息，请阅读“主节点关闭时的情况”。

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，获得群集架构的概述（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（主节点/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

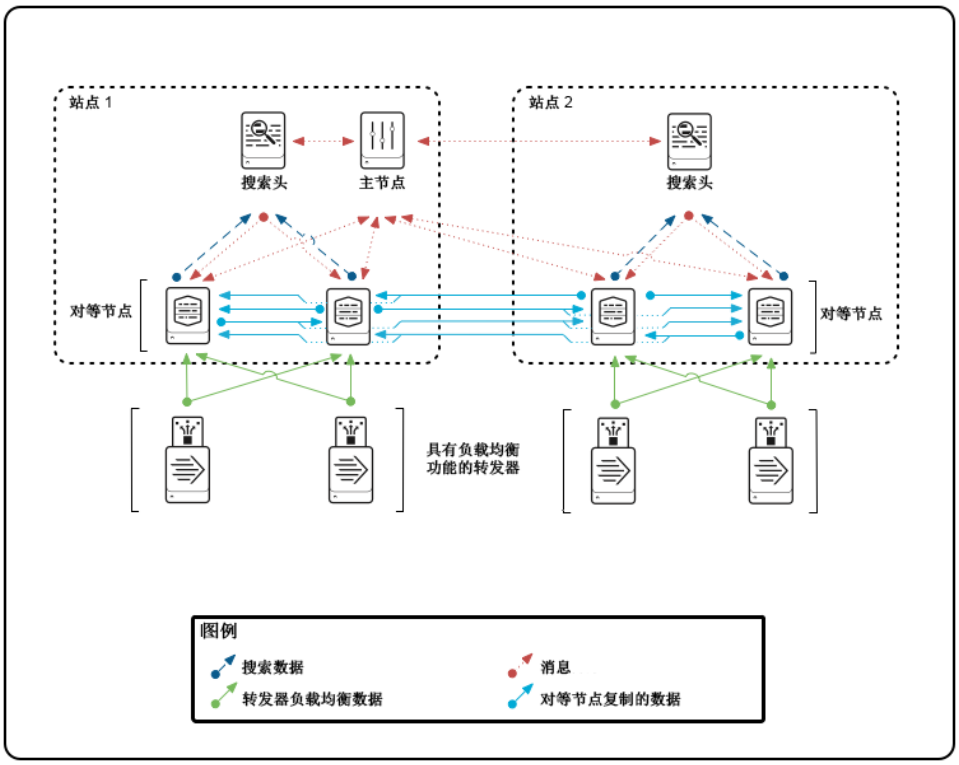
多站点和单个站点节点有如下共同特性：

- 群集有三种类型的节点：主节点、对等节点，以及搜索头。
- 每个群集只有一个主节点。
- 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：

- 主节点控制整个群集。尽管主节点在物理上位于一个站点，主节点实际上却不属于任何一个站点。但是，每个主节点有一个内置搜索头，该搜索头会要求您整体为主节点指定一个站点。注意，主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，请阅读“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，请阅读“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。单个主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，主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执行搜索相关性”。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。

重要提示：阅读本部分之前，您必须理解“保留”数据桶副本的概念。保留的副本是一个等着分配给一个对等节点的副本。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，主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，则群集预留一份非可搜索副本。）

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，群集会预留一份非可搜索副本。

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主节点所在站点，您可启动剩下的一个站点上的备用主节点。请参阅“处理索引器群集主站点故障”。

多站点群集如何处理主节点故障

多站点群集处理主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"主节点关闭时的情况"。

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。

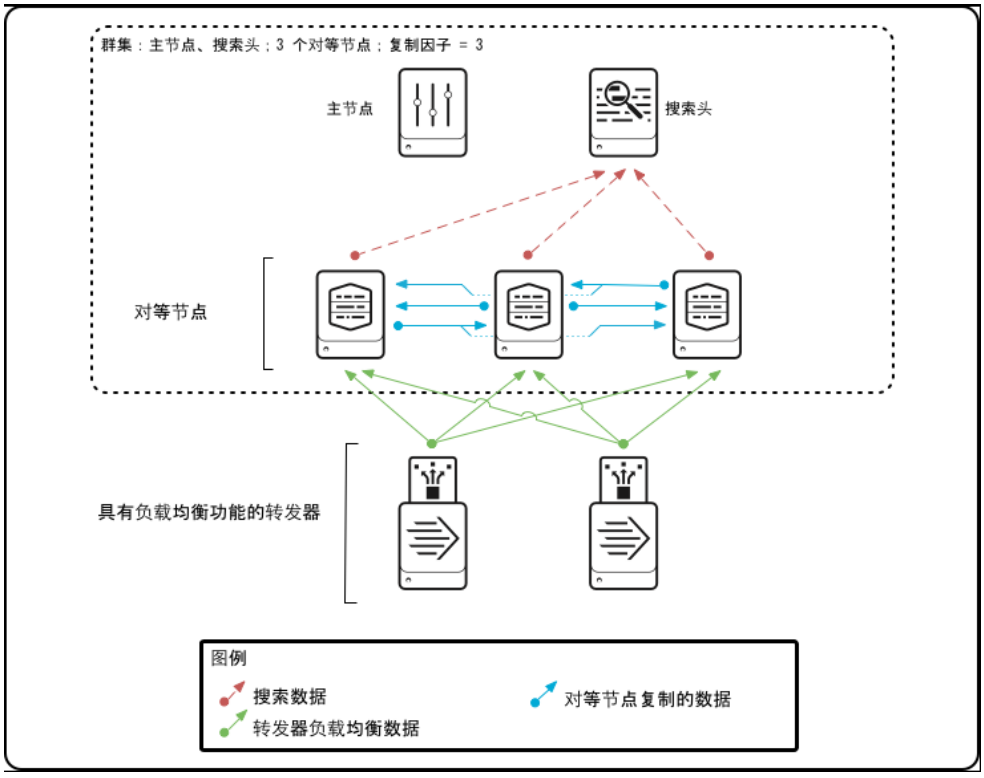
重要提示：在将一个索引器从非群集迁移到群集之前，请确认您的需求。此过程只能单向进行。不支持索引器从群集转换成非群集的过程。

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，可启用并配置群集主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**主节点**，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. 启用主节点。请参阅"启用主节点"。

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.将主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：将主节点数据转发到索引器层"。

其他部署方案

本章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"。
- 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发配置，请使用主题"更新通用对等节点配置"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主节点上，然后以协调的方式分发这些应用到对等节点。

有关如何将应用分发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。主节点、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。

关键要求摘要

以下是需要注意的主要问题：

- 每个群集节点（主、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。
- 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、一个主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。

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，加上一个主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、一个主节点，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"。

重要提示：尽管主节点具有搜索功能，但是您应仅将这些功能用于调试目的。主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主节点部署为生产搜索头。请参阅"主节点的其他角色"。

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：

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

主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主节点不同的版本，但受到下面这些限制：

- 主节点必须运行 6.2 或以上的版本。
- 对等节点和搜索头必须运行 6.1 或以上的版本。
- 主节点运行的版本必须与对等节点和搜索头的相同或更高。

注意：主节点运行的是 6.1 或更早版本时，对等节点和搜索头运行的版本必须与主节点相同。

主节点和 6.1 对等节点之间的兼容性

如果主节点要运行 6.2 或更高版本而对等节点要运行 6.1 版本，您必须把 `use_batch_mask_changes` 属性（位于主节点的 `server.conf` 文件中）设置为 `false`：

```
splunk edit cluster-config -use_batch_mask_changes false
```

如果您使用 CLI 设置此属性，则无需重启主节点。

警告：在所有对等节点都升级到 6.2 或更高版本之后，您必须将 `use_batch_mask_changes` 恢复为 `true`。

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，就同一索引器群集而言，不可以部分对等节点运行 6.n.2 版本而其他对等节点运行 6.n.1 版本。

对等节点和搜索头之间的兼容性

从 6.3 版本开始，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（主节点、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：主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，因为主节点不会建立外部数据索引。

此外，所有群集实例必须运行在同一操作系统上。

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，这意味着主节点、对等节点和搜索头。否则，可能出现各种问题，例如在主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。

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《*安装手册*》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《*安装手册*》中"评估存储要求"中描述的因子。

重要提示：主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《*安装手册*》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 Spunk 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），计算方式如下：
  - 原始数据总计 = (15GB * 3) = 45GB。
  - 总索引文件 = (35GB * 2) = 70 GB。
- **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) = 105 GB。

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：

- 所有群集成员（包括主节点、对等节点和搜索头）都需要位于 Enterprise **许可证池**中，即使它们预期不会建立任何数据索引。
- 群集成员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：

- **在主节点上：**
  - 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发配置，请使用主题"更新通用对等节点配置"中概述的**配置软件包**方法。

有关如何将应用分发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"。

主节点的其他角色

正常情况下，您应该将运行在主节点的 Splunk Enterprise 实例专用于单一目的。然而在受限情况下，主节点实例也能满足一定的其他轻型功能：

- 您可以使用主节点的内置搜索头用于调试目的。
- 您也许可以在主节点上运行一个搜索头群集 Deployer，这取决于主节点的负载。
- 您也许可以在主实例上运行一个监视控制台，具体取决于主实例的负载。

如需在主节点上运行一个 Deployer 或监视控制台，主节点群集应始终低于以下限制：

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

启用索引器群集主节点

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**主节点**。主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：主节点无法作为对等节点或搜索节点承担双重功能。作为主节点启用的 Splunk Enterprise 实例必须只执行该单一索引器群集角色。另外，主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，主节点实例也能处理一些其他轻型功能。请参阅"主节点的其他角色"。

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主节点。

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

- 直接编辑主节点的 `server.conf` 文件。有关详细信息，请参阅"使用 server.conf 配置主节点"。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 配置主节点"。

重要提示：本主题介绍如何仅为单个站点群集启用一个主节点。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"。

启用主节点

作为主节点启用索引器：

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**主节点**并选择**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**。这是验证主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**启用主节点**。

将显示消息，"您必须重新启动 Splunk 以便主节点启动。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：首次启动主节点时，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主节点。如果重启了主节点，对等节点将需要再次重启。

查看主节点仪表板

重新启动后，重新登录到主节点，并返回到 Splunk Web 中的"群集化"页面。此时您会看到主节点群集化仪表板。有关仪表板的信息，请参阅"查看主节点仪表板"。

执行其他配置

请参阅"主节点配置概述"，获得关于部署后主节点配置的信息。

启用对等节点

在阅读此主题之前，请先阅读"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**主节点**，如"启用主节点"所述。首次启动主节点时，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：

- **主节点 URI**。输入主节点 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**。这是验证主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主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"启用主节点"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

在阅读此主题之前，请先阅读"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主节点，如"启用主节点"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：

- **主节点 URI**。输入主节点 URI，包括它的管理端口。例如：`https://10.152.31.202:8089`。
- **安全密钥**。这是验证主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主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

关于在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：将主节点数据转发到索引器层

我们认为最佳的做法是，将所有主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。
- 如果主节点关闭，它将为主节点启用诊断。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，无需在主节点上单独索引。您可通过将主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，除非您已经在主节点上创建了自定义索引器。由于 `_audit` 和 `_internal` 存在于索引器以及主节点上，因此您不需要创建这些索引的单独版本以保存相应的主节点数据。

2.将主节点配置为转发器。在主节点上创建 `outputs.conf` 文件，为跨对等节点集的负载均衡的转发而配置主节点。您也必须关闭主节点上的索引，这样主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 master
[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
autoLB = true
```

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"查看主节点仪表板"。
- 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 实例来充当群集主节点。
- 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（即主节点）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，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主节点的更新，然后主节点会部署对等节点的更新。请参阅"使用部署服务器分发应用到主节点"。

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。将应用存放在主节点上的特殊位置，然后主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，则在主节点上进行更改和添加，然后告诉主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"启用主节点"所述启用主节点。

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主节点的 `$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 `$SPLUNK_HOME/etc/apps` 目录中的应用。
- **c.** 启用索引器作为群集对等节点，如"启用对等节点"所述。
- **d.** 重新启动对等节点，完成启用过程。
- **e.** 验证主节点已将需要的一组应用推送到对等节点的 `$SPLUNK_HOME/etc/slave-apps` 目录。

6.启用所有对等节点后，转到主节点仪表板，并验证正在复制需要的一组索引。此仪表板在"查看主节点仪表板"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

重要提示：本主题的说明假定您正在同时将所有节点升级到新版本。在所有情况下这都没有必要。有关混合版本群集的信息，请参阅"Splunk Enterprise 版本兼容性"。

由于升级性质的不同，升级过程也有所差异。本主题介绍如下情况：

- 从 6.x 升级
- 从 5.x 升级
- 升级到新维护版本（例如，从 6.1.1 到 6.1.2）
- 从 5.0.1 或更早版本升级

从单个站点迁移到多站点？

如果您想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"。

把 6.x 索引器群集升级为更高的 6.x 版本

升级 6.x 索引器群集时（例如，从 6.2 升级到 6.3 或 6.4 等更高的 6.x 群集），您必须让所有群集节点脱机。您无法执行滚动、在线升级。但是，如果您有多站点群集，则可一次升级一个站点。请参阅"多站点索引器群集按站点逐一升级"。

请执行以下步骤：

1.停止主节点。

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.升级主节点，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。**还不要升级节点。**

4.如果主节点还没有运行，则启动主节点，接受所有提示。

5.运行主节点上的 `splunk enable maintenance-mode`。要确认主节点进入了维护模式，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"。

6.升级对等节点和搜索头，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。

7.如果对等节点和搜索头还没有运行，则进行启动。

8.运行主节点上的 `splunk disable maintenance-mode`。要确认主节点退出了维护模式，运行 `splunk show maintenance-mode`。

您可以查看主节点仪表板，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

如果您有多站点群集，可以一次将一个站点升级到新版本。每个站点都有一套完整的主要副本，这使得在升级期间搜索的运行不会中断。

在升级期间，新版本引入的新功能均不可用，直到所有节点均完成升级。

对于双站点群集，升级过程有三个不同的阶段：

1 升级主节点。

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.停止主节点。

2.升级主节点，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.如果主节点还没有运行，则启动主节点，接受所有提示。

4.运行主节点上的 `splunk enable maintenance-mode`。要确认主节点进入了维护模式，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.运行主节点上的 `splunk disable maintenance-mode`。要确认主节点退出了维护模式，运行 `splunk show maintenance-mode`。

9.等待一段时间，直到主节点仪表板上显示搜索因子和复制因子均满足条件。

10.运行主节点上的 `splunk enable maintenance-mode`。要确认主节点进入了维护模式，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.运行主节点上的 `splunk disable maintenance-mode`。要确认主节点退出了维护模式，运行 `splunk show maintenance-mode`。

您可以查看主节点仪表板，以确认所有群集节点已经启用并正在运行。

从 5.x 升级到 6.x

当从 5.x 索引器群集升级到 6.x 群集时，您必须让所有群集节点脱机。您无法执行滚动、在线升级。

请执行以下步骤：

1.在主节点上，运行 `safe_restart_cluster_master` 脚本，并使用 `--get_list` 选项：

```
splunk cmd python safe_restart_cluster_master.py <master_uri> --auth <username>:<password> --get_list
```

注意：对于 `master_uri` 参数，使用 URI 和主节点的端口号。例如：`https://10.152.31.202:8089`

此命令将所有群集数据桶副本及其状态的列表放入文件 `$SPLUNK_HOME/var/run/splunk/cluster/buckets.xml` 中。该列表将在主节点升级后反馈回主节点。

重要提示：要获取本脚本的副本，请从这里复制并粘贴：“safe_restart_cluster_master 脚本”。

有关需要本步骤原因的信息，请参阅“为何需要 safe_restart_cluster_master 脚本”。

2.停止主节点。

3.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`。

4.升级主节点，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。**还不要升级节点。**

5.如果主节点还没有运行，则启动主节点，接受所有提示。

6.运行 `splunk apply cluster-bundle` 命令，使用“更新群集对等节点配置和应用”中介绍的语法。（需要本步骤以避免额外对等节点重新启动，因为 6.0 版本更改了计算配置软件包校验和的方式。）

7.运行主节点上的 `splunk enable maintenance-mode`。要确认主节点进入了维护模式，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”。

8.升级对等节点和搜索头，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。

9.如果对等节点和搜索头还没有运行，则进行启动。

10.在主节点上，再次运行 `safe_restart_cluster_master` 脚本，这次使用 `freeze_from` 选项，指定步骤 1 创建的数据桶列表的位置：

```
splunk cmd python safe_restart_cluster_master.py <master_uri> --auth <username>:<password> --freeze_from
<path_to_buckets_xml>
```

例如：

```
splunk cmd python safe_restart_cluster_master.py <master_uri> --auth admin:changeme --freeze_from
```

`$SPLUNK_HOME/var/run/splunk/cluster/buckets.xml`

这将在步骤 1 获取的冻结数据桶列表导入主节点。

11.运行主节点上的 `splunk disable maintenance-mode`。要确认主节点退出了维护模式，运行 `splunk show maintenance-mode`。

您可以查看主节点仪表板，以确认所有群集节点已经启用并正在运行。

为何需要 safe_restart_cluster_master 脚本

`safe_restart_cluster_master` 脚本以 5.x 主节点处理冻结的数据桶副本的方式处理问题。本问题已在 6.0 版本中得到修复；然而，这在从 5.x 的主升级期间仍是一个问题。本部分提供了该问题的详细信息。

当对等节点冻结数据桶副本时，主节点将停止在该数据桶上进行修复。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。

只要主节点继续运行，这就有用。然而，由于（在 5.x 中）冻结的数据桶的知识未存在于主节点或对等节点上，如果您稍后重新启动主节点，该主节点会将冻结的副本视为丢失的副本（在该数据桶的未冻结副本仍存在于其他对等节点的情况下），同时将执行其常见的修复活动以将群集返回完整状态。如果群集具有许多部分冻结的数据桶，该过程会非常漫长。在该过程完成之前，主节点将无法提交下一次生成。

升级到 6.0 后，为防止您在重新启动主节点时出现这种情况，您必须在主节点上运行 `safe_restart_cluster_master` 脚本。正如在升级程序中介绍的那样，当您最初在 5.x 主节点上使用 `--get_list` 选项运行本脚本时，它将创建所有群集数据桶副本及其状态的列表，包括它们是否被冻结。当在升级主节点到 6.x 后使用 `freeze_from` 选项返回它时，它会将列表导入升级后的主节点，以便不会尝试修复冻结的数据桶。

safe_restart_cluster_master 脚本

要执行升级程序的第 1 至 9 步，您必须运行 `safe_restart_cluster_master` 脚本。**该脚本当前未随产品一起运送。**要获得脚本，直接复制以下列表并保存到`$SPLUNK_HOME/bin/safe_restart_cluster_master.py`。

重要提示：您还必须复制并保存 `parse_xml_v3` 脚本，如下一部分“parse_xml_v3 脚本”中所述。

这里是脚本内容：

```
import httplib2
from urllib import urlencode
import splunk, splunk.rest, splunk.rest.format
from parse_xml_v3 import *
import json
import re
import time
import os
import subprocess

#before restarting the master, store the buckets list in /var/run/splunk/cluster
BUCKET_LIST_PATH = os.path.join(os.environ['SPLUNK_HOME'] , 'var' , 'run' , 'splunk' , 'cluster' , 'buckets.xml')

def get_buckets_list(master_uri, auth):
    f = open(BUCKET_LIST_PATH,'w')
    atom_buckets = get_xml_feed(master_uri +'/services/cluster/master/buckets?count=-1',auth,'GET')
    f.write(atom_buckets)
    f.close()

def change_quiet_period(master_uri, auth):
    args={'quite_period':'600'}
    return get_response_feed(master_uri+'/services/cluster/config/system?quiet_period=600',auth, 'POST')

def num_peers_up(master_uri, auth):
    count = 0
    f= open('peers.xml','w')
    atom_peers = get_xml_feed(master_uri+'/services/cluster/master/peers?count=-1',auth,'GET')
    f.write(atom_peers)
    regex= re.compile('"status">Up')
    f.close()
    file = open('peers.xml','r')
    for line in file:
        match = regex.findall(line)
        for line in match:
            count = count + 1
    file.close()
```

```
    os.remove('peers.xml')
    return count

def wait_for_peers(master_uri,auth,original_number):
    while(num_peers_up(master_uri,auth) != original_number):
        num_peers_not_up = original_number - num_peers_up(master_uri,auth)
        print "Still waiting for " +str(num_peers_not_up) +" peers to join ..."
        time.sleep(5)
    print "All peers have joined"

def get_response_feed(url, auth, method='GET', body=None):
    (user, password) = auth.split(':')
    h = httplib2.Http(disable_ssl_certificate_validation=True)
    h.add_credentials(user, password)
    if body is None:
        body = {}
    response, content = h.request(url, method, urlencode(body))
    if response.status == 401:
        raise Exception("Authorization Failed", url, response)
    elif response.status != 200:
        raise Exception(url, response)

    return splunk.rest.format.parseFeedDocument(content)

def get_xml_feed(url, auth, method='GET', body=None):
    (user, password) = auth.split(':')
    h = httplib2.Http(disable_ssl_certificate_validation=True)
    h.add_credentials(user, password)
    if body is None:
        body = {}
    response, content = h.request(url, method, urlencode(body))
    if response.status == 401:
        raise Exception("Authorization Failed", url, response)
    elif response.status != 200:
        raise Exception(url, response)

    return content

def validate_rest(master_uri, auth):
    return get_response_feed(master_uri + '/services/cluster/master/info', auth)

def freeze_bucket(master_uri, auth, bid):
    return get_response_feed(master_uri + '/services/cluster/master/buckets/' + bid + '/freeze', auth, 'POST')

def freeze_from_file(master_uri,auth,path=BUCKET_LIST_PATH):
    file = open(path) #read the buckets.xml from either path supplied or BUCKET_LIST_PATH

    handler = BucketHandler()
    parse(file, handler)
    buckets = handler.getBuckets()

    fcount = 0
    fdone = 0
    for bid, bucket in buckets.iteritems():
        if bucket.frozen:
            fcount += 1
            try:
                freeze_bucket(master_uri,auth, bid)
                fdone += 1
            except Exception as e:
                print e

    print "Total bucket count:: ", len(buckets), "; number frozen: ", fcount, "; number re-frozen: ", fdone

def restart_master(master_uri,auth):
    change_quiet_period(master_uri,auth)
    original_num_peers = num_peers_up(master_uri,auth)

    print "\n" + "Issuing restart at the master" +"\n"
    subprocess.call([os.path.join(os.environ["SPLUNK_HOME"],"bin","splunk"), "restart"])

    print "\n"+ "Master was restarted" + "\n"
```

```
    print "\n" + "Waiting for all " +str(original_num_peers) + " peers to come back up" +"\n"

    wait_for_peers(master_uri,auth,original_num_peers)

    print "\n" + "Making sure we have the correct number of frozen buckets" + "\n"

if __name__ ==  '__main__':
    usage = "usage: %prog [options] <master_uri> --auth admin:changeme"
    parser = OptionParser(usage)
    parser.add_option("-a","--auth", dest="auth", metavar="user:password", default=':',
                    help="Splunk authentication parameters for the master instance");
    parser.add_option("-g","--get_list", action="store_true",help="get a list of frozen buckets and strore them in
buckets.xml");
    parser.add_option("-f", "--freeze_from",dest="freeze_from",
                    help="path to the file that contains the list of buckets to be frozen. ie path to the
buckets.xml generated by the get_list option above");

    (options, args) = parser.parse_args()

    if len(args) ==  0:
        parser.error("master_uri is required")
    elif len(args) > 1:
        parser.error("incorrect number of arguments")

    master_uri = args[0]
    try:
        validate_rest(master_uri, options.auth)
    except Exception as e:
        print "Failed to access the master info endpoint make sure you've supplied the authentication credentials"
        raise
    # Let's get a list of frozen buckets, stored in
    if(options.get_list):
        print "Only getting the list of buckets and storing it at " + BUCKET_LIST_PATH
        get_buckets_list(master_uri,options.auth)
    elif(options.freeze_from):
        print "Reading the list of buckets from" + options.freeze_from + "and refreezing them"
        freeze_from_file(master_uri,options.auth,options.freeze_from)
    else:
        print "Restarting the master safely to preserve knowledge of frozen buckets"
        get_buckets_list(master_uri,options.auth)
        restart_master(master_uri,options.auth)
        freeze_from_file(master_uri,options.auth,BUCKET_LIST_PATH)
```

parse_xml_v3 脚本

`parse_xml_v3` 脚本包含 `safe_restart_cluster_master` 脚本所需的某些辅助功能。**该脚本当前未随产品一起运送。**要获得脚本，直接复制以下列表并保存到`$SPLUNK_HOME/bin/parse_xml_v3.py`。

这里是脚本内容：

```
import sys
from xml.sax import ContentHandler, parse
from optparse import OptionParser

class PeerBucketFlags:
    def __init__(self):
        self.primary = False
        self.searchable = False

class Bucket:
    def __init__(self):
        self.peer_flags = {}  # key is peer guid
        self.frozen = False

class BucketHandler(ContentHandler):
    def __init__(self):
        ContentHandler.__init__(self)
        self.buckets = {}
        self.in_entry = False
        self.in_peers = False
```

```
    self.save_title = False
    self.save_frozen = False
    self.peer_nesting = 0
    self.current_peer_flags = {}
    self.current_guid = None
    self.current_frozen_flag = ''
    self.current_peer_field = None
    self.current_peer_field_value = ''
    self.current_bucket = ''

def getBuckets(self):
    return self.buckets

def startDocument(self):
    pass

def endDocument(self):
    pass

def startElement(self, name, attrs):
    if name == 'entry':
        self.in_entry = True
    elif self.in_entry and name == 'title':
        self.save_title = True
    elif self.in_entry and name == 's:key' and attrs.get('name') == 'frozen':
        self.save_frozen = True
    elif name == 's:key' and attrs.get('name') == 'peers':
        self.in_peers = True
    elif self.in_peers and name == 's:key':
        self.peer_nesting += 1
        if self.peer_nesting == 1:
            self.current_peer_flags = PeerBucketFlags()
            self.current_guid = attrs.get('name').encode('ascii')
        elif self.peer_nesting == 2:
            self.current_peer_field = attrs.get('name').encode('ascii')
            self.current_peer_field_value = ''

def endElement(self,name):
    if name == 'entry':
        self.in_entry = False
        self.current_bucket=''
    elif self.save_title:
        try:
            (idx, local_id, origin_guid) = self.current_bucket.split('~')
        except ValueError as e:
            print "Invalid? ", self._locator.getLineNumber()
            print self.current_bucket
            print e
            raise
        self.buckets[self.current_bucket] = Bucket()
        self.save_title = False
    elif self.save_frozen:
        if self.current_frozen_flag in [1, '1', 'True', 'true']:
             self.buckets[self.current_bucket].frozen = True
        self.current_frozen_flag = ''
        self.save_frozen = False
    elif self.peer_nesting == 2 and name == 's:key':
        if self.current_peer_field == 'bucket_flags':
            self.current_peer_flags.primary = (self.current_peer_field_value == '0xffffffffffffffff')
        elif self.current_peer_field == 'search_state':
            self.current_peer_flags.searchable = self.current_peer_field_value == 'Searchable'
        # Nesting level goes down in either case.
        self.peer_nesting -= 1
    elif self.peer_nesting == 1 and name == 's:key':
        self.buckets[self.current_bucket].peer_flags[self.current_guid] = self.current_peer_flags
        self.peer_nesting -= 1
    elif self.in_peers and self.peer_nesting == 0 and name == 's:key':
        self.in_peers = False

def characters(self, content):
    if self.save_title:
        self.current_bucket += content.encode('ascii').strip()
```

```
elif self.save_frozen:
    self.current_frozen_flag += content.encode('ascii').strip()
if self.peer_nesting > 0:
    s = content.encode('ascii').strip()
    if s:
        self.current_peer_field_value += s
```

升级到新维护版本

要升级群集到新维护版本（例如，从 6.1 至 6.1.1），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。

重要提示：即便有滚动升级，您还是必须迅速地升级所有节点。群集的正常运行取决于运行在 Splunk Enterprise 同一版本上的所有节点，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。因此，在您准备好升级所有群集节点之前，请不要开始升级过程。如果您升级主节点，但不升级对等节点，群集可能开始生成各种错误和警告。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"。

请执行以下步骤：

1.升级主节点

首先升级主节点。

有关主节点发生故障时会出现什么情况以及主节点恢复后会出现什么情况的信息，请阅读"主节点关闭时的情况"。

2.使主节点进入维护模式

运行主节点上的 `splunk enable maintenance-mode`。要确认主节点进入了维护模式，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"。

3.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 `offline` 命令，如"使对等节点脱机"所述。
- 在升级主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，对能节点升级的站点顺序无关紧要。维护模式没有站点概念。

4.升级搜索头

升级搜索头时对群集的唯一影响是升级期间可能中断搜索。

5.使主节点退出维护模式

运行主节点上的 `splunk disable maintenance-mode`。要确认主节点退出了维护模式，运行 `splunk show maintenance-mode`。

从 5.0.1 或更早版本升级

在从 5.0.1 或更早版本升级到更新版本的过程中，`/cluster` 目录（位于主节点的 `$SPLUNK_HOME/etc/master-apps` 下和对等节点的 `$SPLUNK_HOME/etc/slave-apps` 下）将被重命名为 `/_cluster`。此情形会自动发生。有关此目录的更多详细信息，请参阅"更新通用对等节点配置"。

从 5.0.1 或更早版本升级后，主节点重新启动，然后依次重新启动所有对等节点，推送最新版本的配置软件包（含重命名的 `/_cluster` 目录）。

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。其工作方式类似如下所示：某一对等节点从转发器接收到数据块后，它会在为数据成功建立索引之后向转发器发送一个确认。如果转发器没有从此对等节点收到确认，它就会重新发送数据。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**，每个转发器会查询主节点，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主节点，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。转发器会自动从主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，主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化至其每个目标对等节点。

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。

每个转发器会查询主节点，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，转发器可以查询主节点，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主节点。

2.转发器会按固定的时间间隔轮询主节点，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.主节点将对等节点的 URI 和接收端口传输给转发器。

4.转发器将数据发送给主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，主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：如果主节点发生故障，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主节点发生故障，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，它必须等待主节点返回信息，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.配置主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
```

请参阅*《转发数据手册》*中的“启用接收器”。

注意：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.配置主节点启用索引器发现

在主节点上的 `server.conf` 中，添加该段落：

```
[indexer_discovery]
pass4SymmKey = <string>
polling_rate = <integer>
indexerWeightByDiskCapacity = <bool>
```

请注意以下事项：

- `pass4SymmKey` 属性指定用于主节点和转发器之间通信的安全密钥。对于所有转发器和主节点，该值必须相同。该值可与 `pass4SymmKey` 属性不同，此属性设置于 `[clustering]` 段落之内，用于主节点和群集节点之间的通信。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，来调整转发器轮询主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>
```

请注意以下事项：

- 在 `[indexer_discovery:<name>]` 段落中，`<name>` 可参考 `<name>` 设置（属于 `indexerDiscovery` 属性，位于 `[tcpout:<target_group>]` 段落）。
- `pass4SymmKey` 属性指定用于主节点和转发器之间通信的安全密钥。对于所有转发器和主节点，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>` 是主节点的 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，转发器会从主节点而不是从 `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：

- 主节点启用索引器发现。
- 主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。

在主节点中：`server.conf`：

```
[indexer_discovery]
pass4SymmKey = my_secret
indexerWeightByDiskCapacity = true
```

在每个转发器的 `outputs.conf` 中：

```
[indexer_discovery:master1]
pass4SymmKey = my_secret
master_uri = https://10.152.31.202:8089

[tcpout:group1]
autoLBFrequency = 30
forceTimebasedAutoLB = true
indexerDiscovery = master1
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，可以选择性地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。

重要提示：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。

将此段落添加到转发器的 `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。

警告：如果您为转发器分配了特定的站点且该站点出现故障，此转发器不会将故障转移至其他站点。如果该站点上没有索引器可用，转发器即停止转发数据。

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（位于主节点的 `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主节点之前，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，则索引器会告诉主节点，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主节点，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（在主节点的 `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（位于主节点的 `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.配置主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>
password = <certificate password>
rootCA = <path to certificate authority list>
```

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。

2.配置主节点启用索引器发现

在主节点上的 server.conf 中，添加该段落：

```
[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
sslCertPath = <path to  client certificate>
sslPassword = <CAcert password>
sslRootCAPath = < path to root certificate authority file>
```

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

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）：

```
[tcpout:<peer_target_group>]
useACK=true
```

有关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"。

重要提示：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。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：

- **主节点。**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：

- “使用仪表板配置主节点”
- “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 配置主节点”。
- “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` 属性来配置验证主节点、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`。有关如何在主节点上设置密钥的详细信息，请参阅启用索引器群集主节点。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《*分布式搜索手册*》中的“部署搜索头群集”。

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（主节点、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
```

首次启动主节点时，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。当等待对等节点加入群集时不要重启主节点。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），请阅读"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：

- `master_uri`
- `notify_scan_period`

在搜索头上：

- `master_uri`

在主节点上：

- `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，例如主节点。

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：

- 要启用或编辑主节点，请参阅"使用 CLI 配置主节点"。
- 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 slave -master_uri https://10.160.31.200:8089 -replication_port 9887 -secret
your_key
```

安全密钥验证主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。

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，请在主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
```

配置主节点

主节点配置概述

在启用主节点时已对其进行初始配置，如“启用主节点”中所述。这通常是主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 中的主节点仪表板上编辑配置。请参阅“使用仪表板配置主节点”。
- 可以直接编辑 `[clustering]` 段落（位于主节点的 `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 配置主节点”。
- 您可以使用 CLI。请参阅“使用 CLI 配置主节点”。

在更改主节点的配置之后，需要重新启动主节点以使更改生效。

重要提示：主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。

配置备用主节点

为预防主节点故障，配置备用主节点。这样在当前主节点发生故障时可以接管。请参阅"在索引器群集中替换主节点"。

配置多站点的主节点

多站点主节点与基本的、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"。

使用仪表板配置主节点

您可通过主节点仪表板编辑其配置：

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。
- **主节点配置**。更改这些主节点设置：
  - **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复制因子。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"。

注意：多站点群集的**主节点配置**选项已禁用。

- **分发配置软件包。**分发更新的配置和应用到一组对等节点。请参阅更新群集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：**如果您从群集删除主节点，则整个群集将最终发生故障。

有关使用此仪表板查看群集状态的信息，请参阅查看主节点仪表板。

使用 server.conf 配置主节点

首先阅读

在阅读本主题前，请参阅：

- 使用 server.conf 配置索引器群集。本主题介绍基本的群集配置。它提供关于所有群集节点类型的普遍问题的详细信息。

启用主节点

下面的示例显示当启用一个主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 = master
replication_factor = 4
search_factor = 3
pass4SymmKey = whatever
cluster_label = cluster1
```

该示例指定了以下内容：

- 实例为群集主节点。
- 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”。只在主节点上设置此属性。

重要提示：首次启动主节点时，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主节点。如果重启了主节点，对等节点将需要再次重启。

注意：在 Splunk Web 中启用主节点时，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。

编辑主节点设置

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 配置主节点

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。

启用主节点

下面的示例显示当启用一个主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 master -replication_factor 4 -search_factor 3 -secret your_key -cluster_label
cluster1

splunk restart
```

重要提示：首次启动主节点时，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主节点。如果重启了主节点，对等节点将需要再次重启。

编辑主节点设置

您还可以稍后使用 CLI 来编辑配置。使用 `splunk edit cluster-config` 命令，此命令和最初启用主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：切勿在主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主节点

您可能因为这些原因需要替换主节点：

- 节点故障。
- 需要移动主节点到不同的计算机或站点。

虽然目前没有主节点故障转移操作，但您可以通过配置备用主节点来为主节点故障准备索引器群集。如果主要主节点故障，您可以立即启动备用主节点。您可以使用同样的方法有意地替换主节点。

本主题介绍替换主节点的关键步骤：

1.备份备用/替换主节点需要的文件。

重要提示：这是一个预备步骤。您必须在主节点故障或从系统中删除前执行。

2.准备要连接到新的主节点上的对等节点和搜索头。

3.用新的主节点取代旧的主节点。

重要提示：在多站点群集的情况下，您需要为包括主节点的站点可能出现的故障做准备。请参阅"处理主站点故障"。

备份替换主节点需要的文件

准备替换主节点时，您仅需要复制主节点的静态状态。

注意：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，例如所有群集副本的状态。举例来说，当停机的主节点又回到群集或备用主节点替代了停机的主节点时，群集对等节点会根据需要将此信息传达给主节点。然后主节点将使用此信息重新构建群集动态状态地图。

您需要备份的是两个主节点静态配置，所以您可以稍后将其复制到替换的主节点：

- 主节点的 `server.conf` 文件，这是主节点群集设置的存储位置。您每次更改主节点的群集配置时，都需要备份此文件。
- 主节点的 `$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。每次更新将要推送到对等节点的一组内容时，都需要备份此目录。

确保对等节点和搜索头节点能找到新的主节点

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主节点：

- **替换的主节点使用与主节点相同的 IP 地址和管理端口。** 为确保替换主节点使用相同的 IP 地址，您需要采用基于 DNS 的故障转移、负载均衡器或一些其他技术。在安装过程中设置管理端口，但以后可以通过编辑 `web.conf` 进行更改。
- **替换的主节点不使用与主节点相同的 IP 地址和管理端口。**在这种情况下，在启用新主节点后，您必须更新所有对等节点和搜索头上的 `master_uri` 设置，以指向新主节点的 IP 地址和管理端口。

重要提示：任何一种方法都不需要重新启动对等节点或搜索头节点。

替换主节点

如果您已保存了两套静态配置文件的备份，那么替换主节点的过程就很简单了：

1.停止旧的主节点，如果计划替换它的话。如果因主节点故障进行替换，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。这将是替换的主节点。

3.复制 `sslKeysfilePassword` 设置，将其从替换主节点的 `server.conf` 文件中移到一个临时位置。

4.复制旧主节点的 `server.conf` 和 `$SPLUNK_HOME/etc/master-apps` 文件的备份到替换主节点。

5.删除 `sslKeysfilePassword` 设置（位于复制的 `server.conf` 中），并用在步骤 3 中保存的设置版本替代它。

6.启动替换主节点。

7.确保通过如"确保对等节点和搜索头节点能找到新的主节点"所述的一种方法，使对等节点和搜索头节点指向新的主节点。

注意：如果想跳过步骤 3 和步骤 5，则可以简单地更换属于替换主节点的 `[general]` 和 `[clustering]` 段落，而不是复制整个 `server.conf` 文件。

有关主节点出现故障导致的后果的信息，请阅读"主节点关闭时的情况"。

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，如主节点 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，如主节点以及其用来接收复制数据的端口。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主节点。**当主节点故障时这样做非常有用，但是您得有一个备用主节点准备好接管。关于备用主节点的信息，请参阅"在索引器群集中替换主节点"。
- **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主节点上，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

可通过对等节点群集的仪表板编辑其配置。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。

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：

- 要更改主节点，则编辑**主节点 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**。

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。

使用 server.conf 配置对等节点

首先阅读

在阅读本主题前，请参阅：

- "使用 server.conf 配置索引器群集"。本主题介绍基本的群集配置。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]
master_uri = https://10.152.31.202:8089
mode = slave
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。
- 此对等节点的群集主节点驻留在 `10.152.31.202:8089`。
- 实例为群集对等（"从"）节点。
- 安全密钥为 "whatever"。

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 "slave"。还需要指定 `master_uri` 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 slave -master_uri https://10.160.31.200:8089 -replication_port 9887 -secret
your_key
```

```
splunk restart
```

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，主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.将文件放在主节点上的 `$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`。而是要编辑主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，主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主节点的配置软件包位置。一旦应用在位置中，主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"使用部署服务器分发应用到主节点"。

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。

重要提示：您必须使用主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）。

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.将应用放在主节点上的 `$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，请参阅主题"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，将应用放到相应的常规位置，即，在 `$SPLUNK_HOME/etc/apps` 目录下。

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，可在主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，主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，主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主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.在主节点上编辑一个通用 `indexes.conf` 文件。

2.使用主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：

```
[<newindex>]
repFactor=auto
homePath=<path for hot and warm buckets>
coldPath=<path for cold buckets>
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"。

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，主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。由于您是在主节点上编辑 `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 主节点和一组 Linux 对等节点，编辑文件的 Windows 主节点上指定 `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，而不管主节点和对等节点使用何种操作系统。例如：

```
homePath = $SPLUNK_HOME/var/lib/splunk/defaultdb/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，请单击主节点仪表板上的“索引”选项卡。请参阅“查看主节点仪表板。”

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，从主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，您可以添加文件到主节点上的配置软件包，并指示主节点将文件分发到对等节点。

配置软件包的结构

配置文件包包括一组对所有对等节点通用的文件和应用。

在主节点上

在主节点上，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。本对等节点特定默认 indexes.conf 具有比标准默认 indexes.conf（位于 $SPLUNK_HOME/etc/system/default 下）更高的优先级。
  - 子目录 /_cluster/local 是存放要分发到对等节点的新的或已编辑的配置文件的位置。
  - **对于 5.0/5.0.1 版本升级：**在 Splunk 版本 5.0 和 5.0.1 中，/_cluster 目录命名为 /cluster（无下划线）。将 5.0/5.0.1 主节点升级到 5.0.2 或更高版本时，其 /cluster 目录将自动重命名为 /_cluster。升级完成后，重新启动主节点，主节点在其对等节点上执行滚动重新启动，并使用重命名的 /_cluster 目录将新软件包推送到对等节点。随后，所有对等节点（包括所有 5.0/5.0.1 对等节点）上的 slave-apps 目录将包含重命名的目录。
- /<app-name> 子目录为可选。它们提供将任何应用分发到对等节点的方法。根据需要进行创建和填充。例如，要将 "appBestEver" 分发到对等节点，将该应用的副本放入其自己的子目录中：$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。
- 主节点仅推送 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。

明确指示主节点要在何时将最新配置软件包分发到对等节点。另外，当某一对等节点在主节点中注册时（例如，当该对等节点加入群集时），主节点会将当前配置软件包分发到该对等节点。

警告：主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。主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。对等节点最初从主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主节点上的配置软件包相同：

```
$SPLUNK_HOME/etc/slave-apps/
     _cluster/
          default/
          local/
     <app-name>/
     <app-name>/
     ...
```

重要提示：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。主节点每次重新分发配置软件包时都会覆盖此目录。

Splunk 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”。

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.将文件和应用移入主节点上的配置软件包中。

3.（可选）验证软件包。

4.指示主节点将软件包应用到对等节点。

主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主题。

重要提示：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）都要求执行此操作。详细信息请参阅 indexes.conf repFactor 属性。

2.将文件移动到主节点上

您准备好分发文件和应用时，将它们复制到主节点上的 `$SPLUNK_HOME/etc/master-apps/`：

- 将应用直接放在该目录下。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.验证软件包

本步骤可选。

作为下个步骤的一部分，将数据包应用到对等节点，对等节点会个别验证数据包。每个对等节点会特别验证软件包中所有 `indexes.conf` 文件的设置。所有对等节点都成功地验证软件包后，群集就会完成软件包的应用进程。

当前的步骤中，您可以在不应用软件包的情况下选择性地对其进行验证。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。

验证很有用，例如，若您想确保软件包顺利地应用到所有对等节点，则需要进行验证。验证过程中提供的信息很有用，可用于调试无效的软件包。

如需验证软件包，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。最后，该消息还会建议您运行 `splunk show cluster-bundle-status` 命令查看软件包验证状态。

```
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。

此示例为验证成功后 `splunk show cluster-bundle-status` 返回的内容：

```
master
         active_bundle
                checksum=C249B794C8E0D2F685944C05079767BB
                timestamp=1468607719 (in localtime=Fri Jul 15 11:35:19 2016)
         latest_bundle
                checksum=C249B794C8E0D2F685944C05079767BB
                timestamp=1468607719 (in localtime=Fri Jul 15 11:35:19 2016)
         last_validated_bundle
                checksum=A334B794C8E0D2F685944C05079767BB
                last_validation_succeeded=1
                timestamp=1468607719 (in localtime=Fri Jul 15 11:35:19 2016)
         cluster_status=None

 peer1    4F6107BB-C655-453F-A7A8-2C7651608881         default
         active_bundle=C249B794C8E0D2F685944C05079767BB
         latest_bundle=C249B794C8E0D2F685944C05079767BB
         last_validated_bundle=A334B794C8E0D2F685944C05079767BB
         last_validation_succeeded=1
```

```
        restart_required_apply_bundle=0
        status=Up
...
```

last_validated_bundle 字段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。

检查 cluster_status 字段并查找“验证”状态，即可确定验证是否还在进行中。

此示例为主节点验证失败后 splunk show cluster-bundle-status 返回的内容：

```
master
        active_bundle
               checksum=7CFA6F2EF165481D0F80FCB3BC579885
               timestamp=1468466253 (in localtime=Thu Jul 14 11:17:33 2016)
        latest_bundle
               checksum=7CFA6F2EF165481D0F80FCB3BC579885
               timestamp=1468466253 (in localtime=Thu Jul 14 11:17:33 2016)
        last_validated_bundle
               checksum=CF79637E151B1FDE5D39F3217F31868C
               last_validation_succeeded=0
               timestamp=1468466253 (in localtime=Thu Jul 14 11:17:33 2016)
        invalid_bundle
                              bundle_path : /usr/local/eserv/splunk/var/run/splunk/cluster/remote-
bundle/20f747a0007828a6638fd10bc0d17c44-1468893441.bundle
               bundle_validation_errors_on_master:
                                  checksum : CF79637E151B1FDE5D39F3217F31868C
                                timestamp : 1468893441
        cluster_status=None
               bundle_validation_errors:
                        Cannot create index 'main2': path of homePath must be absolute ('$SPLUK_DB/main2/db')
                        invalid_bundle_id : CF79637E151B1FDE5D39F3217F31868C

peer1   10B02E41-5EDB-487F-9528-5FF1A4FD7311         default
        active_bundle=7CFA6F2EF165481D0F80FCB3BC579885
        latest_bundle=7CFA6F2EF165481D0F80FCB3BC579885
        last_validated_bundle=CF79637E151B1FDE5D39F3217F31868C
        last_validation_succeeded=0
        restart_required_apply_bundle=0
        status=Up
...
```

last_validation_succeeded=0 字段会提示验证失败了。主条目中的信息会提示主节点上的验证失败了。

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。

使用 Splunk Web 应用软件包

要应用配置软件包到对等节点，转到主节点仪表板：

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**。

3.单击仪表板右上角的**编辑**按钮，然后选择**分发配置软件包**选项。

在仪表板上将显示上一次成功推送的信息。它也包含一个**分发配置软件包**按钮。

4.单击**分发配置软件包**按钮。

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。

5.单击**推送更改**以继续。

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。

一旦过程成功完成，对等节点将使用一组新的配置，这些配置现在位于其本地 $SPLUNK_HOME/etc/slave-apps 中。

重要提示：将这些文件保留在 `$SPLUNK_HOME/etc/slave-apps` 中。

有关分发过程内部的详细信息，请阅读通过 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，请在主节点上运行以下 CLI 命令：

```
splunk apply cluster-bundle
```

主节点会发出此警告消息进行响应：

```
Warning: 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，您需要以 `y` 响应消息。通过将 `--answer-yes` 标记附加到该命令，您可以完全避免收到此消息：

```
splunk apply cluster-bundle --answer-yes
```

`splunk apply cluster-bundle` 命令会导致主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，主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，它会向主节点发送消息，主节点会在 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，则必须修复主节点提示的问题并重新运行 `splunk apply cluster-bundle`。

一旦验证完成，主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，请参阅使用滚动重新启动。

当过程完成后，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。

重要提示：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”。

注意： `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。请参阅验证配置软件包。

使用 CLI 查看软件包更新过程的状态

要查看群集软件包更新的进行情况，请从主节点运行以下命令：

```
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。

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。

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，您需要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，对等节点与主节点建立连接，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，则用户需要修复错误并从主节点运行 `splunk apply cluster-bundle`。

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，对等节点仅会加载，避免创建索引或搜索的任何中断。在对等节点上的软件包重新加载阶段确定是否需要重启，并仅在必要时指示主节点启动对等节点的滚动重新启动。

重新加载发生在：

- 在 `props.conf` 中添加新的 sourcetype。
- 添加或更新 `TRANSFORMS-<class>` 段落（位于 `props.conf` 中）。
- 在 `transforms.conf` 中添加新的段落。
- 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。

重新启动发生在：

- 配置软件包包含了对任何配置文件（`indexes.conf`、`props.conf` 或 `transforms.conf` 除外）的更改。
- 除了上面重新加载列表中指定的那些更改之外，对于 `props.conf` 或 `transforms.conf` 做任何更改。
- 执行了确定 indexes.conf 的哪种更改需要重新启动中描述的任何 `indexes.conf` 更改。
- 从配置软件包中删除了现有应用。

使用部署服务器分发应用到主节点

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主节点的配置软件包位置。一旦应用在位置中，主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，来分发应用给主节点。

要使用部署服务器分发文件到主节点上的配置软件包：

1.将主节点配置成部署服务器的客户端，详参*《更新 Splunk Enterprise 实例手册》*中“配置部署客户端”内的介绍。

2.在主节点上，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，以便下载到主节点的配置软件包。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 实例手册》*中的“创建部署应用”。

4.以正常的方式，创建一个或多个将主节点映射到部署应用的服务器类。

5.每个服务器类必须包括 `stateOnClient = noop` 设置：

```
[serverClass:<serverClassName>]
stateOnClient = noop
```

注意：切勿在应用段落级别覆盖该设置。

6.将应用下载到主节点上。

一旦主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。

重要提示：采取步骤确保主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，则确保不会选中**编辑应用**屏幕上的**重新启动 splunkd**。

有关部署服务器的详细信息和如何执行必要的各种操作，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，您可以使用主节点管理通用 `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，当主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主节点，阻止主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"。

关于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主节点。**当主节点发生故障，但是您的该群集拥有可以重定向搜索头到的备用主节点，则这非常有用。关于备用主节点的信息，请参阅"在索引器群集中替换主节点"。
- **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，应该使用主节点将此文件分布到各对等节点。有关如何使用主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"安装知识软件包"章节。

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

您可通过其仪表板编辑搜索头节点配置。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。

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。

- 要更改主节点，则编辑**主节点 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。

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。

注意：**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

首先阅读

在阅读本主题前，请参阅：

- "使用 server.conf 配置索引器群集"。本主题介绍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]
master_uri = https://10.152.31.202:8089
mode = searchhead
pass4SymmKey = whatever
```

该示例指定了以下内容：

- 此搜索头的群集主节点位于 `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。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 `master_uri` 和用于整个群集的安全密钥 (`secret`)：

```
splunk edit cluster-config -mode searchhead -master_uri https://10.160.31.200:8089 -secret your_key

splunk restart
```

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` 命令总是把当前主节点 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主节点，但是它为旧的主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，将搜索头指向主节点。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：

- **主节点 URI**。输入主节点 URI，包括它的管理端口。例如：`https://10.152.31.202:8089`。
- **安全密钥**。这是验证群集主节点、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，此群集的主节点位于 https://10.160.31.200:8089，则运行此命令：

```
splunk add cluster-master https://10.160.31.200:8089
```

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 `master_uri` 属性中指定以逗号分隔的主节点参考列表，后跟每个主节点的各个段落。例如：

```
[clustering]
mode = searchhead
master_uri = clustermaster:east, clustermaster:west

[clustermaster:east]
master_uri=https://SplunkMaster01.example.com:8089
pass4SymmKey=someSecret

[clustermaster:west]
master_uri=https://SplunkMaster02.example.com:8089
pass4SymmKey=anotherSecret
```

在本示例中，搜索头使用 `pass4SymmKey` "someSecret" 与 SplunkMaster01 通信，与 SplunkMaster02 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 `[clustermaster]` 段落下：

```
[general]
site=site1

[clustering]
mode = searchhead
master_uri = clustermaster:multieast, clustermaster:multiwest

[clustermaster:multieast]
multisite=true
master_uri=https://SplunkMaster01.example.com:8089

[clustermaster:multiwest]
multisite=true
master_uri=https://SplunkMaster02.example.com:8089
```

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 `[clustermaster]` 段落下：

```
[clustering]
mode = searchhead
master_uri = clustermaster:multieast, clustermaster:multiwest

[clustermaster:multieast]
multisite=true
master_uri=https://SplunkMaster01.example.com:8089
site=site1

[clustermaster:multiwest]
multisite=true
master_uri=https://SplunkMaster02.example.com:8089
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 `[clustermaster]` 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
master_uri = clustermaster:multi, clustermaster:single

[clustermaster:multi]
multisite=true
master_uri=https://SplunkMaster01.example.com:8089
site=site1

[clustermaster:single]
master_uri=https://SplunkMaster02.example.com:8089
```

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：

- 单个的主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）的一个主节点
- 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：主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，每个主节点有一个内置搜索头，此搜索头需要您在主节点的配置中设置一个站点属性。必须为主节点指定一个站点，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。

在主节点上，应：

- 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` 配置多站点主节点的信息，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 配置多站点主节点的信息，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并把结果返回至搜索头。

如果拥有部分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修复数据桶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利用它。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”。

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（主节点/对等节点/搜索头）：

- `site` 属性指定节点驻留的站点。

在主节点和搜索头上：

- `multisite` 属性指示参与到多站点群集的主节点或搜索头。

仅在主节点上：

- `available_sites` 属性对主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主节点配置中，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 1 到 63 的整数：site1，site2，site3，.....

例如：

```
site=site1
```

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**。

配置主节点

为整个主节点上的群集配置关键属性。下面是主节点的多站点配置示例：

```
[general]
site = site1

[clustering]
mode = master
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。
- 实例为群集主节点。
- 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主节点，但是每个群集仅有一个主节点。
- 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：首次启动主节点时，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"，主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主节点。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]
master_uri = https://10.152.31.202:8089
mode = slave
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。
- 此对等节点的群集主节点位于 `10.152.31.202:8089`。
- 实例为群集对等（“从”）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
master_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。每个主节点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。
- 此搜索头的群集主节点位于 `10.152.31.202:8089`。
- 实例为群集搜索头。
- 安全密钥为 "whatever"。

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：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（主节点、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：首次启动主节点时，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"，主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主节点。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

在主节点上

在更改了下列属性之后，必须重新启动主节点：

- multisite
- available_sites
- site_replication_factor
- site_search_factor

重新启动主节点后，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"。

如果在主节点上更改了 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**。

配置主节点

下面是主节点模式的多站点配置示例：

```
splunk edit cluster-config -mode master -multisite true -available_sites site1,site2 -site site1 -
site_replication_factor origin:2,total:3 -site_search_factor origin:1,total:2

splunk restart
```

该示例指定了以下内容：

- 实例为群集主节点。
- 群集是多站点的。
- 群集包含两个站点：site1 和 site2。
- 主节点位于 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"。

请注意以下事项：

- 每个群集只有一个主节点。
- 必须将多站点群集主节点的 `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`。有关详细信息，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。"

重要提示：首次启动主节点时，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"，主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主节点。如果重启了主节点，对等节点将需要再次重启。

注意：如果您之后更改了主节点的 `site` 值，则不必重新启动主节点。

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 slave -site site1 -master_uri https://10.160.31.200:8089 -replication_port 9887

splunk restart
```

该示例指定了以下内容：

- 实例为群集对等（"从"）节点。
- 实例位于 site1。一个对等节点只能属于一个单个站点。
- 此对等节点的群集主节点位于 `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。

注意：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 -master_uri https://10.160.31.200:8089

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。
- 此搜索头的群集主节点位于 `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
```

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` 命令总是把当前主节点 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。在主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

在主节点的 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 = master
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"。

警告：必须正确配置 `site_replication_factor` 属性。否则，主节点将不会启动。

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=master` 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:8")，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。主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主节点配置中，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。在主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。

在主节点的 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 = master
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，主节点将不会启动。

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=master` 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主节点配置中，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。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：

- 单个站点数据桶（那些在迁移时存在的）继续遵守它们的单个站点复制和搜索因子。不能将它们转换为多站点。
- 多站点数据桶（那些在迁移后创建的）遵循多站点复制和搜索因子规则。

执行多站点迁移

重要提示：迁移过程不会改变正有实例运行的 Splunk Enterprise 的版本。要迁移到多站点群集，实例必须运行于版本 6.1 或更高版本。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。

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，配置多站点的主节点并重启。例如：

```
splunk edit cluster-config -mode master -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。主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.在主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"。

要确认主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主节点和站点。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 slave -site site1 -master_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主节点和站点。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。

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 -master_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. 在主节点上禁用维护模式：

```
splunk disable maintenance-mode
```

要确认主节点退出了维护模式，运行 `splunk show maintenance-mode`。

您可以查看主节点仪表板，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"。

群集如何迁移和维护单个站点数据桶

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。

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。

查看索引器群集状态

查看主节点仪表板

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主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”

访问主节点仪表板

查看主节点的仪表板：

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**。

只能在已作为主节点启用的实例上查看此仪表板。

查看主节点仪表板

主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

如下为打开**对等节点**选项卡时最初显示仪表板的方式：

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“使用仪表板配置主节点”。
- **更多信息。**本按钮提供有关主节点配置的详细信息：
  - **名称。**主节点的 `serverName`（在主节点的 `$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，主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主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。此外，对等节点也不会参与搜索。
    - `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。
    - `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`。主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，直到所有数据桶均完成修复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 Decommissioning 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。**主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。要从主节点列表中删除对等节点，请参阅"从主节点列表中删除对等节点"。

索引选项卡

对于每个索引，主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告为不可搜索。
- **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。
- **累计原始数据大小。**索引的大小，排除热数据桶。

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，主节点仪表板列出：

- **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主节点以获得生成期间信息，则主节点判断此搜索头为关闭。该属性可在 `server.conf` 中配置。

注意：该列表将主节点包含为搜索头之一。尽管主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。主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主节点部署为生产搜索头。同样，与专用搜索头不同，主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过多数据桶副本”。

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主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，使用主节点仪表板，如“查看主节点仪表板”所述。

访问对等节点仪表板

要查看对等节点仪表板：

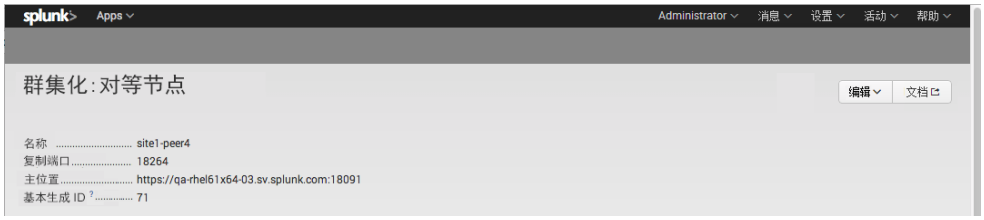
1. 单击 Splunk Web 右上角的**设置**。

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

仪表板类似如下所示：



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。
- **主节点位置。**主节点的 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

搜索头仪表板类似如下所示：

仪表板列出搜索头属于的所有群集的主节点，以及每个群集状态的一些信息。

有关主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，它复制了主节点仪表板中的信息，如"查看主节点仪表板"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

使对等节点脱机

使用 CLI 命令 `splunk offline` 使对等节点脱机。使用 `offline` 命令，可以将搜索中断的可能性降至最低。

取决于您的需求，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。完整群集还是有效的群集。

当您使对等节点暂时脱机时

当您使对等节点暂时脱机时，该节点通常会进行升级或进行短时间的其他维护作业。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较短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，主节点会启动操作让其返回到有效状态，但通常不会让它返回到完整状态，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，主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。

脱机命令

CLI 命令 `offline` 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，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。这样便从根本上消除对现有或将来搜索的干扰。`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，取决于命令的运行方式。

重要提示：为使脱机命令按预期执行，搜索因子至少必须为 2。原因是：搜索因子为 2 时，始终有数据的备用可搜索副本。如果具有可搜索数据的对等节点进入脱机状态，新的搜索可以立即切换到该数据的备用可搜索副本，从而将不可完全搜索群集的时间降至最低。另一方面，如果搜索因子仅为 1，主节点必须先使不可搜索副本成为可搜索副本，才能再次对全部数据运行搜索。这可能需要相当长一段时间，详参评估对等节点取消配置时群集的恢复时间中的介绍。

`offline` 命令有两个版本：

- **`splunk offline`**：这是 `offline` 命令的快速版，主要用于使对等节点暂时脱机。对等节点最多在 5-10 分钟后关闭，即使搜索仍在进行。可使用这个版本的命令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。
- **`splunk offline --enforce-counts`**：这是命令的 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。若调用了 `enforce-counts` 标记，则对等节点在所有搜索和补救活动均已完成后才会关闭。

重要提示：停用对等节点时，使用 `offline` 命令，而非 `stop` 命令。`offline` 命令会停止对等节点，但执行此操作时对搜索的干扰最小。

暂时停用对等节点：快速版的脱机命令

以下是快速版 `offline` 命令的语法：

```
splunk offline
```

直接在对等节点上运行此命令。

运行此命令时，在主节点完成必需活动的协调，使群集恢复到有效状态，最多需要 5-10 分钟，在这之后，对等节点关闭。对等节点在关闭之前会继续参与正在进行的搜索。

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，并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，则主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，由此来延长主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主节点的 `restart_timeout` 时间（默认设置为 60 秒），则可更改该设置的值。对主节点运行此 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，您不需重新启动主节点。

还可以在主节点的 `server.conf` 中更改此值。

永远停用对等节点：enforce-counts 版的脱机命令

以下是 enforce-counts 版 `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 **decommissioning** 的过程，此在过程中主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。

注意：如果群集无法返回完整状态，则对等节点无法关闭。这通常是因为群集上的剩余对等节点数小于复制因子。例如，群集的复制因子是 3 而且只有 3 个对等节点，那么如果要使其中一个节点脱机，则主节点无法返回到完整状态。如果一定要在这种情况下使一个对等节点脱机，则要改用快速版的 `offline` 命令。

对等节点在取消配置之前会继续参与正在进行的搜索。

如果稍后重新启动此对等节点，则会将其添加回原来的群集中。

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"，但它会继续出现在主节点仪表板的对等节点列表中。要从主节点列表中删除对等节点，请参阅从主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，主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，则主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**。（这些索引的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

热数据桶复制期间，会生成某些条件，并导致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。

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。

警告：在群集处于维护模式时，主节点将强制执行复制因子或搜索因子策略。在维护模式期间出现的唯一数据桶修复是主节点在必要时尝试重新分配主要给可用可搜索数据桶副本。因此，群集可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**。

注意：CLI 命令 `apply cluster-bundle` 和 `rolling-restart` 默认结合维护模式功能到它们的行为，因此不必在这些命令正在运行时进行显式调用。当调用这些操作时，表面维护模式打开的消息将显示在主节点仪表板上。

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，在主节点上运行本 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。

维护模式在主节点重新启动期间不会持续。

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。返回值 `0` 表示维护模式已关闭。

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。

重新启动主节点或对等节点时，主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。如果更改了主节点配置，只要重启主节点。如果更新了一组通用的对等节点配置，主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主节点和对等节点：

1. 重新启动主节点，方法和任何实例一样。例如，在主节点上运行以下 CLI 命令：

```
splunk restart
```

2. 主节点重新启动后，请等待所有对等节点在主节点重新注册，并且主节点仪表板指示所有对等节点和索引均可搜索。请参阅查看主节点仪表板。

3. 在主节点上运行以下 CLI 命令，作为一个组重新启动对等节点：

```
splunk rolling-restart cluster-peers
```

请参阅使用滚动重新启动。

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，主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： 主节点等待的实际时间取决于 server.conf 中主节点的 `restart_timeout` 属性值。此属性默认为 60 秒。如果需要主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，主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，主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。主节点等待的实际时间由主节点的 `heartbeat_timeout` 属性决定。未经咨询，建议您不要更改 60 秒的默认值。

使用滚动重新启动

`splunk rolling-restart` 命令将执行所有对等节点的分阶段重新启动，以便整个群集可以在重新启动过程中继续执行其功能。

在以下情况下会发生滚动重新启动：

- 通过调用 `splunk rolling-restart` 命令可启动滚动重新启动。
- 主节点启动滚动重新启动。在分发配置软件包到对等节点后，主节点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。

警告：除非绝对必要，否则不要调用 `splunk rolling-restart` 命令。重启对等节点集会导致长时间大量的数据桶修复。

滚动重新启动如何工作

滚动重新启动的运行方式是：主节点一次发送重新启动消息给约 10%（默认）的对等节点。如果群集中的对等节点少于 10 个，则它会一次对一个对等节点发起重新启动。

这些对等节点重新启动并与主节点联系之后，主节点会向另一个 10% 的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。此方法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。

在滚动重新启动结束后，主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。

以下是关于滚动重新启动的行为要注意的几件事情：

- 主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，无法保证群集将完全可搜索。

指定滚动重新启动

从主节点调用 `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，主节点将一次发出重新启动命令给 10% 的对等节点。然而，该百分比可以通过 `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` 配置本属性。例如，要更改重新启动行为以便主节点一次重新启动 20% 的对等节点，运行本命令：

```
splunk edit cluster-config -percent_peers_to_restart 20
```

要使主节点一次重新启动所有对等节点，运行命令时采用值 `100`：

```
splunk edit cluster-config -percent_peers_to_restart 100
```

这在某些情况下有用，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` 属性后，您仍需要运行 `splunk rolling-restart` 命令以真正开始重新启动。

在多站点群集上滚动重新启动

在多站点群集中，您可以指定滚动重新启动进行站点识别。即，主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。

默认情况下，滚动重新启动过程不进行站点识别。主节点重新启动对等节点时无需考虑对等节点驻留在何处。

在多站点群集上调用滚动重新启动

当您为多站点群集调用 `splunk rolling-restart` 命令时，您可以指定以下行为：

- 通过 `-site-by-site` 参数指定重新启动是否应以站点识别的方式继续进行。

- 通过 `-site-order` 参数指定站点顺序。

以下是命令的多站点版本：

```
splunk rolling-restart cluster-peers [-site-by-site true|false] [-site-order site<n>,site<n>, ...]
```

请注意以下事项：

- `-site-by-site` 参数：
  - 该参数指定重新启动时是否进行站点识别。
  - 默认为 false，即，主节点从整个群集间随机选择每轮的对等节点，无需考虑它们驻留在哪个站点。
- `-site-order` 参数：
  - 该参数指定站点重新启动的顺序。
  - 当使用该选项时您必须列出所有可用站点。
  - 只有 `-site-by-site` 参数设为 true，该参数才有意义。
  - 默认情况下，如果未指定该参数，则随机选择站点。

例如，假设您有一个三站点群集：

```
splunk rolling-restart cluster-peers -site-by-site true -site-order site1,site3,site2
```

`-site-by-site` 的值为 true 意味着主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进行。`-site-order` 参数使得主节点按以下顺序：site1、site3、site2 来执行重新启动。因此，主节点先在 site1 上执行滚动重新启动并一直等到它完成，然后在 site3 上执行滚动重新启动并一直等到它完成，然后在 site2 上执行滚动重新启动。

主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

您可以使用单个站点群集中所用的相同方法，通过编辑 `percent_peers_to_restart` 属性（位于 `server.conf` 中）来指定同时重新启动的对等节点的百分比。该百分比一直是在全局范围内进行计算，即使对于站点识别滚动重新启动也是如此。因此，假定默认值为 10%，在有两个站点的群集中，site1 上有 10 个对等节点，site2 上有 20 个对等节点，总共有 30 个对等节点，则主节点会一次重新启动三个对等节点。

如果重新启动不进行站点识别，则主节点会随机地从任意一个站点选择三个对等节点。对于任何一轮特定的重新启动，它可以从一个站点选择两个对等节点，并从另一个站点选择一个对等节点，或者从单个站点选择全部三个对等节点。

反之，如果重新启动会进行站点识别，则会如下所示进行重新启动：

1. 首先主节点选择一个要重新启动的站点，例如 site2。（可配置站点顺序。）

2. 主节点重新启动 site2 上的三个对等节点。

3. 主节点重新启动 site2 上的另外三个对等节点，依此类推，直到它重新启动 site2 上的全部 20 个对等节点。在 site2 上的最后一轮重新启动，主节点只会重新启动两个对等节点，因为它不会将该轮重新启动拆分到多个站点间进行。

4. 主节点重新启动 site1 上的三个对等节点。

5. 主节点重新启动 site1 上的另外三个对等节点，依此类推，直到它重新启动 site1 上的全部 10 个对等节点。在 site1 上的最后一轮重新启动，它只会重新启动一个对等节点，因为此时只剩下一个对等节点。

重新平衡索引器群集

重新平衡索引器群集是在重新平衡群集上的数据桶副本。索引器群集重新平衡分两种：

- **主要副本重新平衡。** 主要副本重新平衡的目标是平衡所有对等节点上的搜索负载。实现这个目标的做法是重新分布主要副本，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。
- **数据重新平衡。** 数据重新平衡的目标是平衡群集中所有可用对等节点上的存储分布。实现这个目标的做法是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。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。数据重新平衡将平衡可搜索、不可搜索的副本和主要副本。由于平衡对象包括可搜索副本，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。

重新平衡索引器群集的主要数据桶副本

当启动或重新启动主节点或对等节点时，主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，主节点将在必要时，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。

只要对等节点或主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。

注意：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本副本。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，主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主节点上的 `/services/cluster/master/control/control/rebalance_primaries` REST 端点。例如，在主节点上运行本命令：

```
curl -k -u admin:pass --request POST \
  https://localhost:8089/services/cluster/master/control/control/rebalance_primaries
```

有关详细信息，请参阅有关群集/主节点/控制的 REST API 文档。

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

为了解任意站点上的主要负载，您可以使用 `cluster/master/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。

请参阅群集/主节点/对等节点的 REST API 文档。

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。
- 主节点重新加入群集。
- 您手动按主节点上的 `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

一些因素可能会造成数据副本的分布失衡。其中包括：

- **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。

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。

除磁盘使用考量外，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。

数据重新平衡如何工作

主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，主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。

请注意这些关键方面的重新平衡：

- 数据重新平衡将平衡不可搜索和可搜索数据桶副本，以实现这两类数据桶副本的均衡分布。
- 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。

重要提示：数据重新平衡将在重新平衡进程开始时和之后删除任何过多的数据桶副本，因为重新平衡进程中会生成多余的数据桶副本。

如想深入了解重新平衡进程，请参阅数据重新平衡进程。

在多站点群集中，主节点会在站点配置允许的情况下，先于各站点上平衡数据桶副本。之后，主节点才会在每个站点内平衡数据桶副本。请参阅多站点群集中的数据重新平衡。

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

启动群集数据重新平衡

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，请在主节点上运行以下 CLI 命令：

```
splunk rebalance cluster-data -action start [-index index_name] [-max_runtime interval_in_minutes]
```

请注意以下事项：

- 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群集数据重新平衡。

您也可以从主节点仪表板上启动重新平衡。请参阅使用主节点仪表板启动和配置重新平衡。

注意：最好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。

停止群集数据重新平衡

如需提前停止数据重新平衡，请在主节点上运行以下 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，请在主节点上运行以下 CLI 命令：

```
splunk rebalance cluster-data -action status
```

您也可以使用主节点仪表板查看重新平衡状态。请参阅使用主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

主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（位于主节点的 `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 master -rebalance_threshold 0.95 -auth admin:changeme
```

您也可以通过主节点仪表板配置重新平衡阈值。请参阅使用主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，主节点会执行必须要的重新平衡，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。

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。

限制每次重新平衡进程可以处理的数据桶数量上限的属性也决定着所有修复活动的对等节点负载：`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）。

使用主节点仪表板启动和配置重新平衡

您也可以通过主节点仪表板启动和配置重新平衡。请参阅使用仪表板配置主节点。

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。
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。

数据重新平衡的限制

数据重新平衡对存储利用率和并发搜索的影响是有限的：

- **存储利用率。**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，该平衡进程会实现存储的最佳近似平衡。具体来说：
  - 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。
  - 所有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，数据重新平衡进程将停止。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。
- **并发搜索。**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。因为这个原因，最佳做法是在维护窗口中运行数据重新平衡。

数据重新平衡进程

以下为数据重新平衡在单站点群集上如何工作的进一步详细描述。

1. 主节点从群集中删除所有多余的数据桶副本。数据桶不可搜索或可搜索副本的数量超过复制因子和搜索因子指定的数量时，群集中就会出现多余的副本。

2. 主节点把所有数据桶添加到修复列表。如果主节点仅平衡一个索引，它只会把该索引的数据桶添加到修复列表。

3. 主节点会为群集内每个对等节点上的所有索引计算超出/低于数据桶副本平均值的数量。计算结果将决定哪些对等节点上一个或多个索引的副本数量超出平均值，以及哪些对等节点上的副本数量低于平均值。重新平衡进程将把数据桶副本从副本数量超出平均值的对等节点移动到副本数量不足的对等节点。该操作将以索引为基础逐个执行。

例如，若一个群集上有两个索引 index1 和 index2，主节点将计算每个对等节点上的数据桶副本是高于还是低于副本平均值。两个索引的副本计算将单独进行。假设 peer1 上 index1 的数据桶副本数量高于平均值，而 index2 的数据桶副本数量则低于平均值。重新平衡进程会从 peer1 上删除 index1 数据桶副本，从而修正 index1 副本数量过多的问题。该进程还会把 index2 数据桶副本添加到 peer1，从而修正 index2 副本数量不足的问题。因此，一个对等节点在重新平衡操作期间既可能获得数据桶副本也可能会丢失数据桶副本。

4. 主节点会根据修复列表依次处理每个数据桶。

A. 如果主节点发现某对等节点上当前数据桶索引的数据桶副本数量高于平均值而且还拥有当前数据桶的不可搜索副本，则（如果存在这样的节点）该主节点将指示群集在当前数据桶索引的数据桶副本总数最少且尚未拥有此数据桶副本的对等节点上制作此数据桶的一个副本。该主节点仅为此数据桶的一个不可搜索副本执行此操作。

B. 如果找到一个满足相同条件集的数据桶可搜索副本，主节点将为此副本重复执行该进程。

C. 此时，如果群集制作了步骤 4a 和 4b 中的不可搜索和可搜索副本，则群集中最多可能会有两个多余的数据桶副本。如果有多余的副本，主节点将从索引的各数据桶副本总数最多且正在处理一个数据桶副本的对等节点上删除一个多余的数据桶副本。如果还有另一个多余的数据桶副本，主节点接下来将从副本总数次多且正在处理一个数据桶副本的对等节点上删除该多余的数据桶副本。

D. 主节点将重新为每个对等节点上的所有索引计算超出/低于数据桶平均值的数量。

E. 主节点将把处理过的数据桶移动到修复列表末尾。如果需要在所有数据桶的第一次迭代之后继续重新平衡进程，

主节点将对数据桶执行第二次处理。

5.主节点将继续逐个处理列表中的所有数据桶，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。限制可同时进行重新平衡处理的数据桶数量上限的属性也决定着所有修复活动的对等节点负载：`max_peer_rep_load` 和 `max_peer_build_load` 位于 `[clustering]` 段落中，该段落属于 `server.conf`。

注意：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）。

6.所有对等节点上各索引的副本数量落入 `rebalance_threshold` 值确定的平均值范围内后，数据重新平衡就会停止。有两种方法可以提前终止重新平衡进程：手动干预或达到重新平衡启动时指定的可选时间限制。

7.主要节点将对群集内的所有数据桶执行主要副本重新平衡。

多站点群集中的数据重新平衡

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。

重新平衡多站点群集的进程遵照重新平衡单站点群集时使用的基本逻辑。但是，为调节站点间和站内平衡，重新平衡进程在每个数据桶内都分两个阶段来操作：

1.站点间。主节点平衡各站点上的数据桶：

A. 主节点将确定站点复制因子（之后是站点搜索因子）是否拥有非显式数据桶副本，亦即，未绑定至某个站点的数据桶。

B. 如果有含非显式副本的数据桶，主节点将为其寻找一个符合以下条件的对等节点：拥有该数据桶的副本，且节点上的数据桶数量高于平均值。与单站点重新平衡一样，主节点以索引为基础逐个做出决定。

C. 主节点把数据桶从步骤 1b 中识别出的对等节点复制到另一个站点上数据桶数量低于平均值的对等节点；前提是另一个站点中需存在这样的对等节点。

D. 平衡数据桶的不可搜索和可搜索副本后，若有一个或多个多余的副本，主节点将从整个群集中持有数据桶副本数量最多且拥有该数据桶副本的对等节点（如果有多个多余的副本，对应的对等节点也有多个）上删除多余的副本。

2.站内。主节点平衡各站点内的数据桶：

A. 主节点会针对每个站点进行检查，以查看含数据桶副本的站点上是否有对等节点持有高于平均值的数据桶副本。与单站点重新平衡一样，主节点以索引为基础逐个做出决定。

B 如果在步骤 2a 中识别到对等节点，主节点将把数据桶副本复制到同一站点上副本数量低于平均值且尚未拥有该数据桶副本的对等节点,前提是站点中存在这样的对等节点。

C. 主节点会先针对不可搜索副本然后再针对可搜索副本执行重新平衡，步骤与单站点群集中的重新平衡相同。

D. 如果在上一个步骤中产生了多余的副本，主节点会将其删除。

和在单站点中的重新平衡一样，主节点会按顺序处理每个数据桶。亦即，每个数据桶的站内阶段将紧随其站点间阶段执行。

从索引器群集中删除过多数据桶副本

当对等节点被关闭，但在随后返回索引器群集时，对等节点在关闭时保留的任何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。

您可以通过主节点仪表板或 CLI 查看和删除过多数据桶副本。

使用主节点仪表板

查看或删除过多数据桶副本：

1.在主节点上，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**。

这会打开主节点仪表板。

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，从主节点运行本命令：

```
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，从主节点运行本命令：

```
splunk remove excess-buckets [index-name]
```

主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。

从主节点列表中删除对等节点

在对等节点关闭后，它仍然在主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），但它仍会出现在主节点仪表板上。

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，主节点将终止整个操作。

您也可以通过重启主节点从主节点列表中删除对等节点。

有关主节点仪表板的对等节点列表的信息，请参阅查看主节点仪表板。

管理多站点索引器群集

处理主站点故障

如果发生故障的站点是主节点所在站点，您可启动剩下的一个站点上的新的主节点。同时，群集会继续以最佳状态工作。对等节点继续根据目标对等节点列表流出数据给其他节点，这些节点在主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。

应在至少一个其他站点上配置一个备用主节点，以准备可能发生的主节点故障。

执行以下事项：

1. 为至少一个非当前主节点所在站点配置一个备用主节点。关于执行此操作的详细信息，请参阅在索引器群集中替换主节点。**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主节点，详参在索引器群集中替换主节点中的介绍。

3. 遵照主节点重新启动或站点故障之后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。

新的主节点现已作为旧的主节点的完全替代。

注意： 如果站点之后恢复了，您需要将该站点上的节点指向新的主节点。请参阅确保对等节点和搜索头节点能找到新的主节点。

主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

当主节点重新启动时，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主节点。
- 带有主节点的站点故障，您在另一个站点启动备用主节点。

如果一个站点故障，但是运行在另一个站点的主节点仍在工作，建立索引照常继续，因为主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，在主节点上运行 `splunk set indexing-ready` 命令以开启建立索引：

```
splunk set indexing-ready -auth admin:changeme
```

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"，其中主节点位于 site1。如果 site2 故障，您随后重新启动主节点，主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，如果有主节点的 site1 故障，您在 site2 上启动一个备用主节点，则新的主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主节点去重新启动建立索引。

重要提示： 在所列情况下，每次重新启动主节点时都必须运行 `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（主节点/对等节点/搜索头）。

2. 在主节点上，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. 启动主节点。

6. 启动对等节点和搜索头。

请注意以下事项：

- 主节点忽略留在 `server.conf` 中的任何多站点属性（`site_replication_factor` 等等）。
- 重新启动后，主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。主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

若要取消站点配置，您需要停止站点上的对等节点或把对等节点重新分配给剩余的站点。您还需要重新配置多个站点特定属性。

重新配置属性

取消一个站点的属性时，您需要更改主节点上 `server.conf` 内的多个站特定属性：

- `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。当前的主题中有此属性的深入介绍。请参阅映射取消了配置的站点。

更改上述任意属性后，您必须重新启动主节点。

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。这样做会导致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

在您实际开始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的详细过程。

如需把已取消配置的站点映射到剩余站点，请使用 `site_mappings` 属性（位于 `server.conf` 中）。您只能在主节点上设置该属性，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主节点。

示例

为简化说明，所有示例均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。

取消站点配置的详细过程

此过程将取消多站点索引器群集中一个站点的配置。

继续之前，请注意以下重要事项：

- 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站点群集，则该站点为单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主节点，主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。
- 主节点不得位于您想要取消其配置的站点上。如果主节点刚好位于您想要取消其配置的站点上，请遵照处理主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其配置的站点上。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其配置的站点上。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，您必须等到主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. 如果站点中有搜索头，请禁用该搜索头或更改其 `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. 如果有转发器使用了指定已取消配置站点的索引器方法，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. 运行主节点上的 `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. 要确认主节点进入了维护模式，运行 `splunk show maintenance-mode`。
5. 在主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. 重新启动主节点。重启之后，所做的属性更改将生效。此步骤还会从维护模式中删除主节点，并为剩余站点上的对等节点启动修复活动。

   注意：主节点重新启动后，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. 在已取消配置站点的所有对等节点上运行 `splunk stop`。
8. 如需验证取消配置是否成功，请查看主节点仪表板的顶部。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。请参阅查看主节点仪表板。

   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。

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。

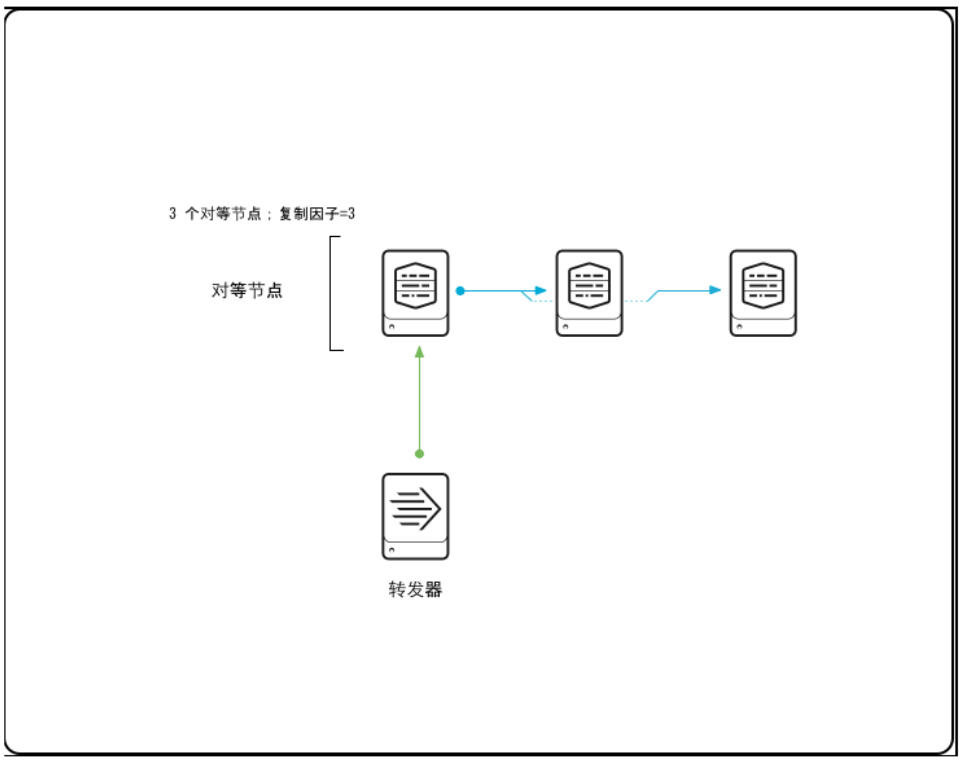
增加复制因子可以增加系统容许的对等节点故障数量。复制因子为 2 时，只能容许一个节点故障；复制因子为 3 时，可以容许两个并发故障，依此类推。

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（“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

配置主节点时，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"。

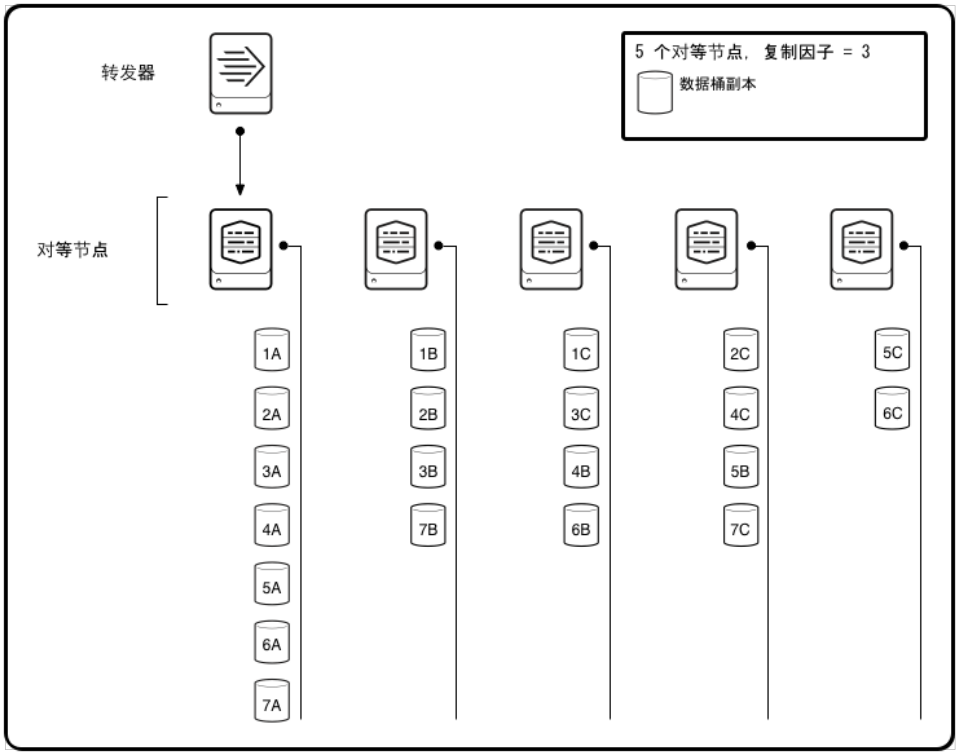
数据桶和索引器群集

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（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）。每次来源对等节点启动新的热数据桶时，主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**）

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，源对等节点流出它的原始数据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的相对大小的信息，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。

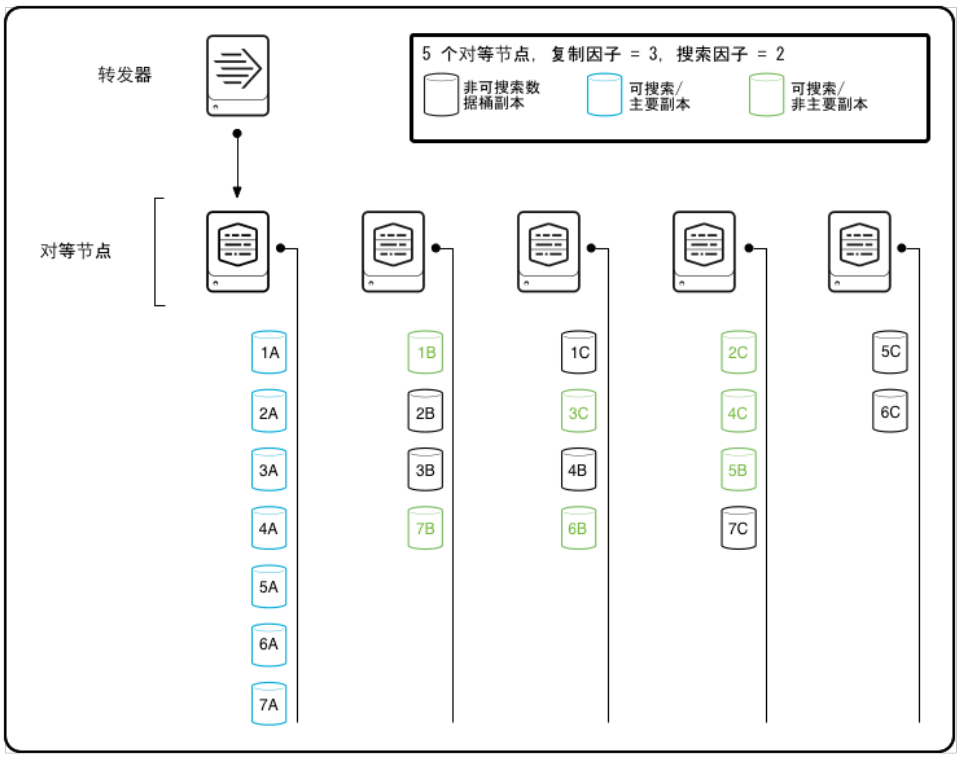
最初，数据源自的对等节点上的数据桶的副本是主要副本，但可随时间变更。例如，如果节点故障，主节点将主要性从已关闭的对等节点上的主要副本重新分配给剩余对等节点上的相应的可搜索副本。有关此过程的更多信息，请阅读"对等节点故障时的情况"。

当主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。
- 主节点重新加入群集。
- 您手动按主节点上的 `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，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。主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。

当前在主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，主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：

- 主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性信息。
- 对于每个搜索，搜索头会使用从主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：

- 主节点联机。
- 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，主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，如当您手动按主节点上的 `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"。
- 主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主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

搜索头会定期轮询主节点以获取最新的生成期间信息。当生成期间更改时，主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。

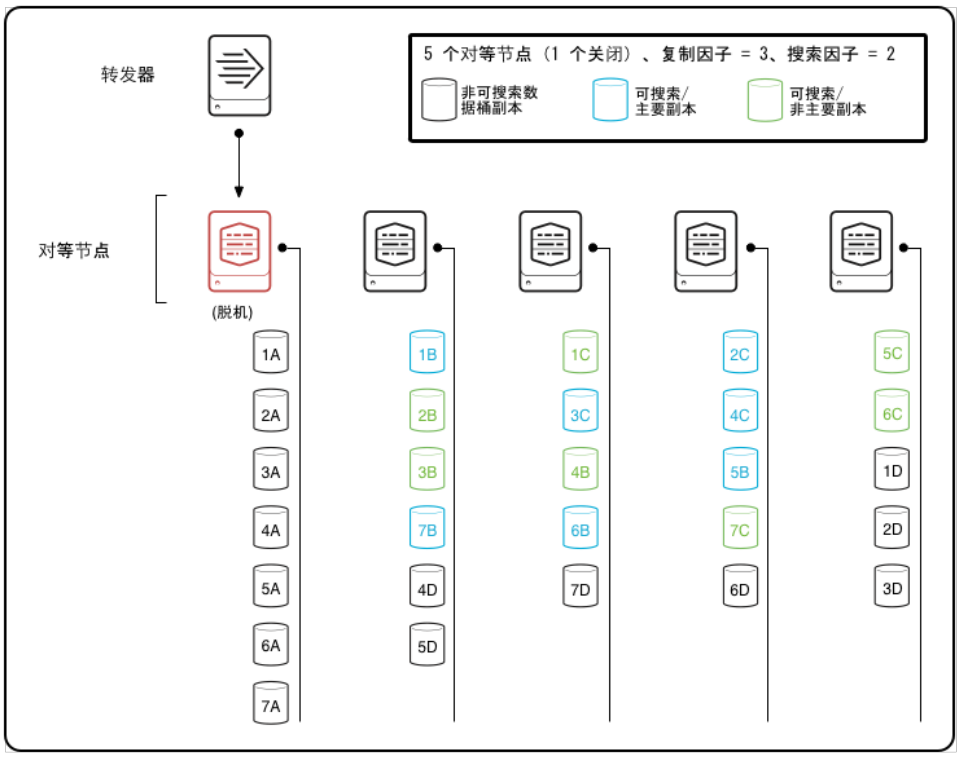
通常，搜索在主要数据桶副本的最近生成期间内发生。但是，如果是长时间运行的搜索，搜索可能在早期生成期间内运行。这种情况通常会发生，因为对等节点在搜索期间发生故障。这样，即使某些数据缺失（由于对等节点故障），长时间运行的搜索也能够完成。另一种方法是重新启动搜索，如有必要，您始终可以手动这样做。

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主节点创建新的生成期间的原因在于，对等节点发生故障时，主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来源节点已发生故障，并且主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，主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，它会通知主节点。主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该数据桶的所有副本，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大小上限，导致最旧的数据桶遭冻结，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。

- 完整群集还是有效的群集，但是有效的群集不一定完整。

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，主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“当某对等节点故障时会发生什么情况”。

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：

- **来源节点。**来源节点从转发器或其他外部来源获取数据。
- **目标对等节点。**目标对等节点从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。主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

当源对等节点启动一个热数据桶时，主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。

主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.该对等节点向主节点注册并从主节点接收最新的**配置软件包**。

2.主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.主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，则主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，主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，如果复制因子为 2，则主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主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 中主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：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。主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. 来源对等节点将其已经滚动数据桶的情况通知给主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，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主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. 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**将从主节点获取**搜索对等节点**的列表。此外，还会从主节点获取生成期间 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主节点（换言之，已启动并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. 该搜索头将与主节点接触。

2. 主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（即当前已注册到主节点的对等节点）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。

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，搜索只在请求的搜索头所在站点上的节点之间运行。

在多站点群集中总是启用搜索相关性。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。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，仅有本地节点会响应。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。

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。

如果本地站点上的一些对等节点有故障，站点因此没有完全补足主要副本，则远程节点将参与搜索，提供来自本站点缺失的所有主要副本的结果。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。

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。

您可以配置主节点，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。**建议采用这种方式。**

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。

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，则必须在主节点的 `server.conf` 文件中设置此属性：

```
[clustering]
summary_replication = true
```

必须重新启动主节点。

也可以在主节点上使用 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（位于主节点的 `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：

- 主节点启动时
- 对等节点加入新群集时
- 对等节点加入现有群集时

主节点启动时

当主节点联机（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主节点注册，主节点会将其添加到群集中。主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，您必须首先启用主节点，然后启用对等节点，如“索引器群集部署概述”所述。主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。

后续重新启动主节点时，它会安静地等待 60 秒钟，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，主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，在您重新启动主节点时，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。

在 60 秒时间结束后，可以查看主节点仪表板了解群集状态的信息。

有关主节点发生故障并重新启动时发生事件的更多信息，请参阅“主节点关闭时的情况”。

对等节点加入新群集时

当您首次部署群集时，必须首先启用主节点，然后启用对等节点，如“索引器群集部署概述”中所述。主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主节点注册，主节点会将最新的**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主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主节点进行注册，主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，因为主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，但主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

对等节点可以有意地（通过调用 CLI 命令 `offline`，如“使对等节点脱机”中所述）或无意地（例如，由于服务器崩溃）关闭。

无论对等节点的故障原因为何，主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**。主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，主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，主节点通常可以重新创建有效的群集。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。主节点是否还能够重新创建完整的群集，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：

- 主节点滚动**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

`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。

`offline` 命令有两个版本：

- `splunk offline`：这是 `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 5-10 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，则对等节点在所有搜索和补救活动均已完成后才会关闭。

有关如何使对等节点脱机的基础知识，请阅读主题"使对等节点脱机"。

快速版的脱机命令

快速版 `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。主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。主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 5-10 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。它不再向主节点发送检测信号。

5.重新启动计数。对等节点关闭后，主节点将按照 `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，则主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，主节点将执行补救操作来修复群集数据桶。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 `offline` 命令的语法如下：

```
splunk offline --enforce-counts
```

此命令版本启动一个名为 **decommissioning** 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。主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。主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。主节点将执行操作来修复群集数据桶。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。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。对等节点在以下条件满足时关闭：1）所有执行中的搜索均已完成；2）群集返回到完整状态。一旦关闭，此对等节点无法再向主节点发送检测信号。此时，对等节点的状态变为 `GracefulShutdown`。

注意：在步骤 4 中，如果群集无法返回完整状态，则对等节点无法关闭。这通常是因为剩余对等节点的数量小于复制因子。

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，它会停止向主节点发送定期的检测信号。这样，主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，主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，主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，主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。

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主节点如何 1) 重新创建有效而又完整的群集，以及如何 2) 在保留的活动节点不足时创建有效但并不完整的群集。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而又完整的群集

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，主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，主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，主节点会确定 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，主节点会确定 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但不完整的群集

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，主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，主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所有文件。）

2. 然后，主节点指示来自步骤 1 的节点标记 5 个最新的可搜索副本为主要。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。

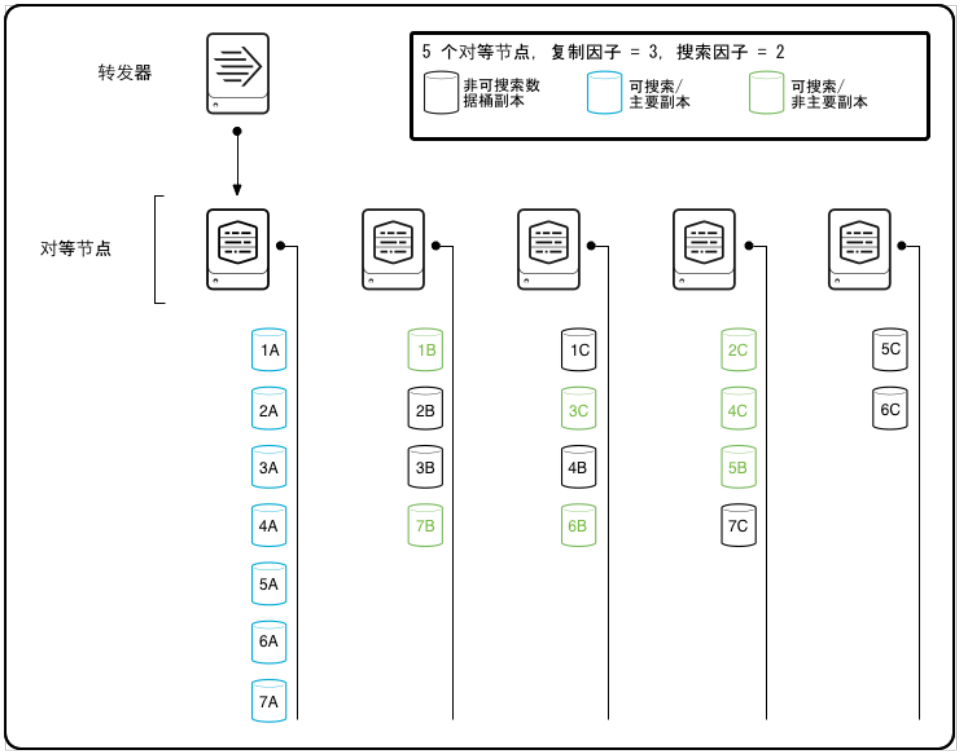
一旦步骤 2 完成，群集重新获得有效状态。任何随后的搜索将提供完整的结果。

3. 对于故障节点上的总计 20 个数据桶副本，主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。

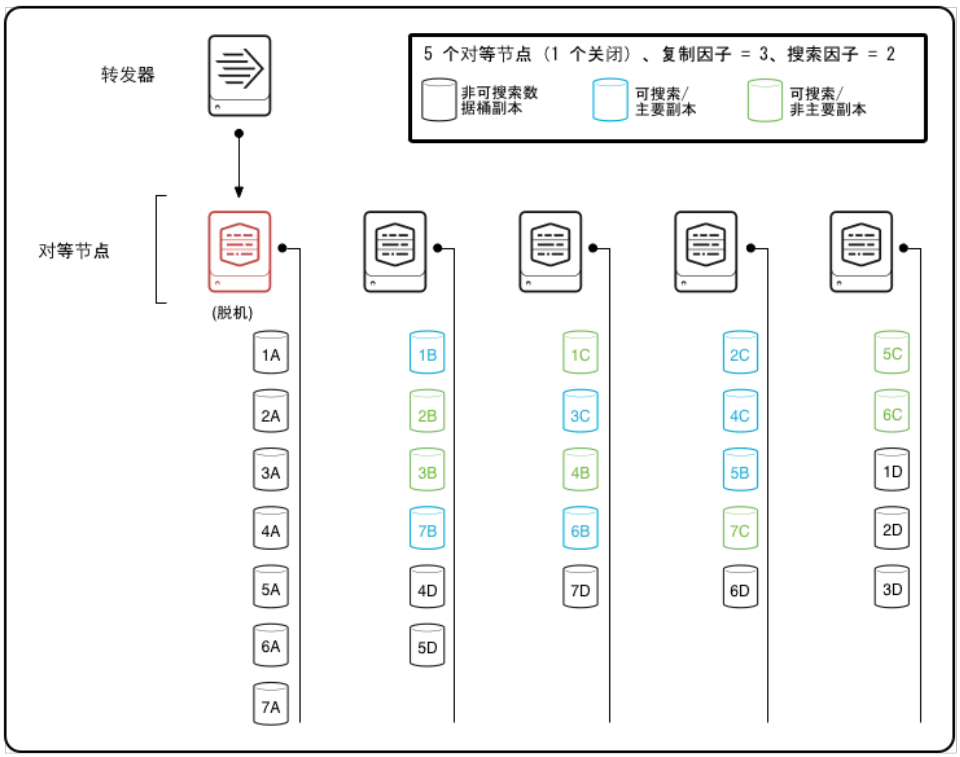
图片

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来源节点发生故障后，主节点已经发出了让剩余对等节点修复数据桶的指令：



主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. 主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

您可以在主节点仪表板上查看数据桶修复状态。请参阅"查看主节点仪表板。"

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，它会开始向主节点发送检测信号。主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，则主节点会将这些副本添加到它所维护的数据桶计数中。主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： 当对等节点与主节点连接时，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。

主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，您必须首先了解主节点如何跟踪数据桶副本。

主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。

主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，主节点会为每个站点（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，主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，一个有效的、完整的群集有：

- 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，主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，主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，主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。主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。

删除额外数据桶副本

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。有关详细信息，请阅读“从索引器群集中删除过多数据桶副本”主题。

主节点关闭时的情况

主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，如果主节点发生故障，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，您仍然应将主节点故障视为严重故障。

为了应付可能出现的主节点故障，可以配置备用主节点，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主节点”。

当主节点发生故障时

如果主节点发生故障，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，它通常会联系主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，如果对等节点在主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，而主节点仍旧故障，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主节点，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。）搜索头将使用它在主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。

主节点重新启动时

对等节点继续无限期地发送检测信号，以便主节点重新启动时，它们可以检测到主节点并重新连接。

当主节点重新启动时，它会安静地等待 60 秒钟，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，主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主节点注册，主节点会发起任何需要的数据桶修复活动，以确保群集**有效完整**。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。

在 60 秒时间结束后，可以查看主节点仪表板了解群集状态的准确信息。

注意：在您重新启动主节点时，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。

故障排除索引器和索引器群集

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，可能无法对某些最近接收的数据进行搜索。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，针对所有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）。

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。

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
```

`fsck repair` 命令可能需要数小时运行，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"中的描述，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
```

注意：不能使用 `splunk fsck` 命令去修复单个数据桶。而要使用 `splunk rebuild` 命令。

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>
```

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。

运行 `rebuild` 命令之前，必须停止索引器。

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。

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

需要您重新构建索引级清单的情况非常罕见，但如果您需要这样做，可以通过索引器提供的以下几个命令来完成：

警告：只有在获得 Splunk 支持引导的情况下才能使用这些命令。不要自行重新构建清单。

两个索引级清单文件是 .bucketManifest 和 .metaManifest。.bucketManifest 文件包含索引中所有数据桶的列表。比如，您手动将数据桶复制到某索引中时，可能需要重新构建此清单。.metaManifest 文件包含已分发到索引级元数据文件中的数据桶的列表。

以下命令**仅**为主索引重新构建 .bucketManifest 和 .metaManifest 文件以及 homePath 中的所有 *.data 文件。而**不**重新构建个别数据桶的元数据：

```
splunk _internal call /data/indexes/main/rebuild-metadata-and-manifests
```

如果只想重新构建 .metaManifest 和 homePath/*.data 文件，改用以下命令：

```
splunk _internal call /data/indexes/main/rebuild-metadata
```

如果只想重新构建 .bucketManifest 文件，使用以下命令：

```
splunk _internal call /data/indexes/main/rebuild-bucket-manifest
```

可以使用星号 (*) 通配符来重新构建所有索引的清单。例如：

```
splunk _internal call /data/indexes/*/rebuild-metadata-and-manifests
```

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原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，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，以下横幅消息将在主节点的仪表板中显示一次或多次，这取决于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。

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。

源对等节点上的复制故障的证据

复制失败证据显示在源对等节点的 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，因为主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，这条消息会出现在主节点仪表板上。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。

处理异常数据桶的问题

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主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。

警告：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. 从主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. 选择“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，您可以采取下列补救措施：

1. 为您想要管理的数据桶选择“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。

警告：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

主节点上有一数据桶所有副本的信息。但在某些情况下，主节点提供的一特定对等节点上的副本信息可能是错误的。主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问题。

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主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，因此若对等节点和主节点就某个数据桶副本提供了不同的状态信息，则主节点上的信息是错误的。

如需解决这个问题，请重新同步主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主节点。

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。

警告：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主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软件包 (>200MB)，软件包验证可能由于各种超时而失败。要修复的话：

1.编辑主节点的 `server.conf` 文件来包含这些设置：

```
[sslConfig]
allowSslCompression = false

[clustering]
heartbeat_timeout = 600
```

2.重新启动主节点。

3.将包含一些更新的 `server.conf` 设置的软件包推送给对等节点：

a. 将上一次成功推送后添加到软件包的应用移动到某个临时位置。

b. 在主节点上创建采用以下设置的 `$SPLUNK_HOME/etc/master-apps/_cluster/local/server.conf`：

```
[general]
useHTTPClientCompression = true

[clustering]
heartbeat_period = 30
cxn_timeout = 300
send_timeout = 300
rcv_timeout = 300
```

c. 将软件包推送到对等节点：

```
splunk apply cluster-bundle
```

这会启动所有对等节点的滚动重新启动。

4.将在步骤 3a 中从软件包里删除的应用重新添加到软件包中，同时保留新的 `server.conf` 文件。

5.将扩展软件包推送到对等节点。

使用 Hadoop Data Roll 归档数据

有关使用 Hadoop Data Roll 归档索引

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。

该功能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。

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 1.6 及以上的 Java 版本。不过，为了获取最佳结果，我们建议您升级至 1.6 以上的版本。

以下几种 Hadoop 分布和版本支持 Hadoop Data Roll：

- Apache Hadoop
  - 0.20
  - 1.0.2
  - 1.0.3
  - 1.0.4
  - 2.4
  - 2.6
  - 2.7
- Cloudera Distribution Including Apache Hadoop
  - 4
  - 4.2
  - 4.3.0
  - 4.4 （HA NN 和 HA JT）
  - 5.0
  - 5.3
  - 5.3 (HA)
  - 5.4
  - 5.5
  - 5.6
- Hortonworks Data Platform (HDP)
  - 1.3
  - 2.0
  - 2.1
  - 2.2
  - 2.3
  - 2.4
- MapR
  - 2.1
  - 3.0
  - 5.0
- Amazon Elastic MapReduce (EMR)
- IBM InfoSphere BigInsights
  - 5.1
- Pivotal HD

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 Ana lytics 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/`

归档如何工作

一旦您将索引配置为归档：

1.`splunk_archiver` 应用使用软件包复制把您的配置信息分布到所有相关的索引器。

2.一个小时后，Hadoop Data Roll 每隔 17 分钟自动运行一次 `| archivebuckets` 命令，该命令将在所有索引器上启动归档进程。

3.Hadoop Data Roll 把冷和温的数据桶数据从 Splunk Enterprise 索引器复制到支持 Hadoop 的文件系统，如 HDFS 或 S3。

4.归档的数据桶已在 Splunk Web 内准备就绪，可随时执行搜索。

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。使用搜索命令可搜索存储于 Hadoop 中的索引，有关搜索命令的信息，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。

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。

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 中的一个目录，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

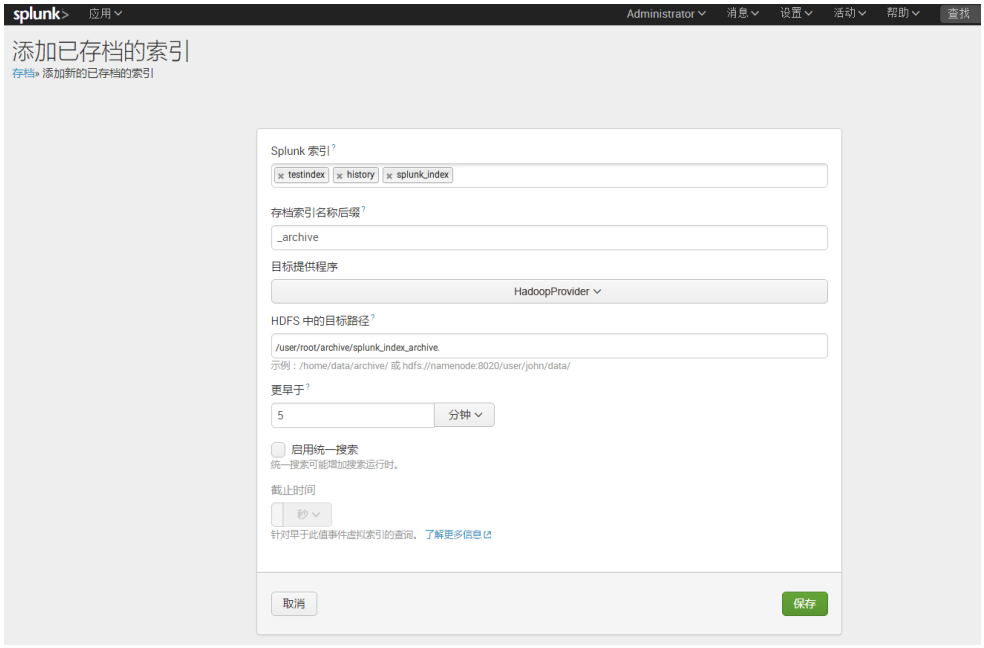
开始之前请先注意以下事项：

- 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

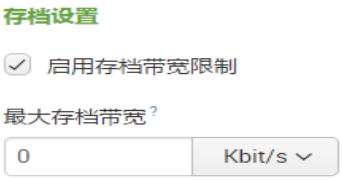
若您担心持续归档所需的带宽，您可以设置带宽限制。您为提供程序设置限制后，该限制将被应用到分配至该提供程序的所有索引。

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 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，您就可以创建报表，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，您既可以创建虚拟索引报表也可以创建本地索引报表。更多有关创建报表的信息，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创建单份报表时，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 的搜索语言创建报表。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 目录内区分挑选删除项目的顺序，Splunk Analytics for Hadoop 将根据修改时间或文件顺序挑选要删除的项目。）

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头和归档索引上。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创建各种搜索，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 `coldToFrozenSh.script`，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"
```