About indexer clusters and index replication
About indexer clusters and index replication

**Indexer clusters** are groups of Splunk Enterprise indexers configured to replicate each others’ data, so that the system keeps multiple copies of all data. This process is known as **index replication**. By maintaining multiple, identical copies of Splunk Enterprise data, clusters prevent data loss while promoting data availability for searching.

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

The key benefits of index replication are:

- **Data availability.** An indexer is always available to handle incoming data, and the indexed data is available for searching.
- **Data fidelity.** You never lose any data. You have assurance that the data sent to the cluster is exactly the same data that gets stored in the cluster and that a search can later access.
- **Data recovery.** Your system can tolerate downed indexers without losing data or losing access to data.
- **Disaster recovery.** With multisite clustering, your system can tolerate the failure of an entire data center.
- **Search affinity.** With multisite clustering, search heads can access the entire set of data through their local sites, greatly reducing long-distance network traffic.

The key trade-off in index replication is between the benefits of data availability/recovery and the costs of storage (and, to a minor degree, increased processing load). The degree of data recovery that the cluster possesses is directly proportional to the number of copies of data it maintains. But maintaining more copies of data means higher storage requirements. To manage this trade-off to match the needs of your enterprise, you can configure the number of copies of data that you want the cluster to maintain. This is known as the **replication factor**.

**Note:** Clusters with **SmartStore** indexes rely on the inherent capabilities of a remote object store to ensure high availability, data fidelity, data recovery, and disaster recovery for most of the index data. See About SmartStore.

You can also use clusters to scale indexing capacity, even in situations where index replication is not a requirement. See "Use indexer clusters to scale indexing".
Note: Search head clusters provide high availability and scalability for groups of search heads. They are a separate feature from indexer clusters, but you can combine them with indexer clusters to build a high availability, scalable solution across your entire Splunk Enterprise deployment. See "About search head clustering" in the Distributed Search manual.

Parts of an indexer cluster

An indexer cluster is a group of Splunk Enterprise instances, or nodes, that, working in concert, provide a redundant indexing and searching capability. Each cluster has three types of nodes:

- A single master node to manage the cluster.
- Several to many peer nodes to index and maintain multiple copies of the data and to search the data.
- One or more search heads to coordinate searches across the set of peer nodes.

The master node manages the cluster. It coordinates the replicating activities of the peer nodes and tells the search head where to find data. It also helps manage the configuration of peer nodes and orchestrates remedial activities if a peer goes down.

The peer nodes receive and index incoming data, just like non-clustered, stand-alone indexers. Unlike stand-alone indexers, however, peer nodes also replicate data from other nodes in the cluster. A peer node can index its own incoming data while simultaneously storing copies of data from other nodes. You must have at least as many peer nodes as the replication factor. That is, to support a replication factor of 3, you need a minimum of three peer nodes.

The search head runs searches across the set of peer nodes. You must use a search head to manage searches across indexer clusters.

For most purposes, it is recommended that you use forwarders to get data into the cluster.

Here is a diagram of a basic, single-site indexer cluster, containing three peer nodes and supporting a replication factor of 3:
This diagram shows a simple deployment, similar to a small-scale non-clustered deployment, with some forwarders sending load-balanced data to a group of indexers (peer nodes), and the indexers sending search results to a search head. There are two additions that you don't find in a non-clustered deployment:

- The indexers are streaming copies of their data to other indexers.
- The master node, while it doesn't participate in any data streaming, coordinates a range of activities involving the search peers and the search head.

**Multisite indexer clusters**

**Multisite indexer clusters** allow you to maintain complete copies of your indexed data in multiple locations. This offers the advantages of enhanced disaster recovery and search affinity. You can specify the number of copies of data on each site. Multisite clusters are similar in most respects to basic, single-site clusters, with some differences in configuration and behavior. See "Multisite indexer clusters".

**How to set up a cluster**

Clusters are easy to set up. The process is similar to setting up a group of stand-alone indexers. Basically, you install the indexers and perform a bit of configuration.
The main difference is that you also need to identify and enable the cluster nodes. You designate one Splunk Enterprise instance as the master node and other instances as peer nodes. You need at least as many peer nodes as the size of your replication factor. To increase indexing capacity for horizontal scaling, you just add more peer nodes.

You also need to set up one or more search heads to manage searches across the peers and to consolidate the results for the user.

You enable cluster nodes in the same way that you configure any settings in Splunk Enterprise: through Splunk Web or the CLI, or directly, by editing configuration files.

See the chapter "Deploy the indexer cluster".

**How to search a cluster**

You search a cluster the same way you search any non-clustered group of indexers. You submit your searches through a search head.

What happens behind the scenes is a bit different, though. Once you have submitted your search, the search head consults the master node to determine the current set of peer nodes. The search head then distributes the search tasks directly to those peers. The peers do their part and send their results back to the search head, which then consolidates the results and returns them to Splunk Web. From the user’s standpoint, it is no different than searching any standalone indexer or non-clustered group of indexers. See "How search works in an indexer cluster."

With a multisite cluster, you can also implement search affinity. In search affinity, a search head gets search results only from indexers local to its site, when possible. At the same time, the search still has access to the full set of data. See "Implement search affinity in a multisite indexer cluster."

**Before you go any further**

Clusters are easy to set up and use, but you need to have a good grounding in the basics of Splunk Enterprise indexing and deployment first. Before you continue, make sure you know this stuff:

- **How to configure indexers.** See "How the indexer stores indexes", along with the other topics in the current manual that describe managing indexes.
• **What a search head does.** For an introduction to distributed search and search heads, see "About distributed search" in the *Distributed Search* manual.

• **How to use a forwarder to get data into an indexer.** See "Use forwarders" in the *Getting Data In* manual.

**Migrating from a non-clustered Splunk Enterprise deployment?**

Clustered indexers have several different requirements from non-clustered indexers. It is important that you be aware of these issues before you migrate your indexers. For details, see "Key differences between clustered and non-clustered Splunk Enterprise deployments of indexers". After you read that material, go to "Migrate non-clustered indexers to a clustered environment" for details on the actual migration process.