Splunk® Enterprise Managing Indexers and Clusters of Indexers 7.3.0

Use forwarders to get data into the indexer cluster

Generated: 6/21/2019 6:01 am
Use forwarders to get data into the indexer cluster

The main reasons to use forwarders with indexer clusters are:

- **To ensure that all incoming data gets indexed.** By activating the forwarder's optional **indexer acknowledgment** feature, you can ensure that all incoming data gets indexed and stored on the cluster. See "How indexer acknowledgment works."

- **To handle potential node failure.** With **load-balanced** forwarders, if one peer in the group goes down, the forwarder continues to send its data to the remaining peers in the group. See "How load balancing works."

- **To simplify the process of connecting data sources and peer nodes.** By enabling indexer discovery on your forwarders, the forwarders automatically load balance across all available peer nodes, including any that are later added to the cluster. See "Advantages of the indexer discovery method."

To use forwarders to get data into clusters, you must perform two types of configuration:

- Connect forwarders to peer nodes.
- Configure the forwarders' data inputs.

Before continuing, you must be familiar with forwarders and how to use them to get data into Splunk Enterprise. For an introduction to forwarders, read "About forwarding and receiving" in the *Forwarding Data* manual. Subsequent topics in that manual describe all aspects of deploying and configuring forwarders.

**Connect forwarders to peer nodes**

There are two ways to connect forwarders to peer nodes:

- **Use the indexer discovery feature.** With **indexer discovery**, each forwarder queries the master node for a list of all peer nodes in the cluster. It then uses load balancing to forward data to the set of peer nodes. In the case of a multisite cluster, a forwarder can optionally query the master for a list of all peers on a single site. See "Use indexer discovery to connect forwarders to peer nodes."
• **Connect forwarders directly to peer nodes.** This is the traditional method for establishing forwarder/indexer connectivity. You specify the peer nodes directly on the forwarders as **receivers**. See "Connect forwarders directly to peer nodes."

### Advantages of the indexer discovery method

Indexer discovery has advantages over the traditional method:

- When new peer nodes join the cluster, you do not need to reconfigure and restart your forwarders to connect to the new peers. The forwarder automatically gets the updated list of peers from the master. It uses load balancing to forward to all peers in the list.

- You can add new forwarders without needing to determine the current set of cluster peers. You just configure indexer discovery on the new forwarders.

- You can use **weighted load balancing** when forwarding data across the set of peers. With indexer discovery, the master can track the amount of total disk space on each peer and communicate that information to the forwarders. The forwarders then adjust the amount of data they send to each peer, based on the disk capacity.

### Configure the data inputs to each forwarder

After you specify the connection between the forwarders and the receiving peers using the method you prefer, you must specify the data inputs to each forwarder, so that the forwarder has data to send to the cluster. You usually do this by editing the forwarder's **inputs.conf** file.

Read the *Getting Data In* manual, starting with "What Splunk can index" for detailed information on configuring data inputs. The topic in that manual entitled "Use forwarders" provides an introduction to specifying data inputs on forwarders.

### How indexer acknowledgment works

To ensure end-to-end data fidelity, you must explicitly enable indexer acknowledgment on each forwarder sending data to the cluster.

In brief, indexer acknowledgment works like this: The forwarder sends data
continuously to the receiving peer, in blocks of approximately 64kB. The forwarder maintains a copy of each block in memory until it gets an acknowledgment from the peer. While waiting, it continues to send more data blocks.

If all goes well, the receiving peer:

1. receives the block of data, parses and indexes it, and writes the data (raw data and index data) to the file system.

2. streams copies of the raw data to each of its target peers.

3. receives notification from each target peer of either a successful or unsuccessful write.

4. sends an acknowledgment back to the forwarder.

The acknowledgment assures the forwarder that the data was successfully written to the cluster. Upon receiving the acknowledgment, the forwarder releases the block from memory.

If the forwarder does not receive the acknowledgment, that means there was a failure along the way. Either the receiving peer went down or that peer was unable to contact its set of target peers. The forwarder then automatically resends the block of data. If the forwarder is using load-balancing, it sends the block to another receiving node in the load-balanced group. If the forwarder is not set up for load-balancing, it attempts to resend data to the same node as before.

For more information on how indexer acknowledgment works, read "Protect against loss of in-flight data" in the Forwarding Data manual.

**How load balancing works**

In load balancing, the forwarder distributes incoming data across several receiving peer nodes. Each node gets a portion of the total data, and together the receiving nodes get all the data.

Splunk forwarders perform "automatic load balancing". The forwarder routes data to different nodes based on a specified time interval. For example, assume you have a load-balanced group consisting of three peer nodes: A, B, and C. At the interval specified by the `autoLBFrequency` attribute in `outputs.conf` (30 seconds by default), the forwarder switches the data stream to another node in the group, selected at random. So, every 30 seconds, the forwarder might switch from node
B to node A to node C, and so on. If one node is down, the forwarder immediately switches to another.

**Note:** To expand on this a bit, each of the forwarder’s inputs has its own data stream. At the specified interval, the forwarder switches the data stream to the newly selected node, if it is safe to do so. If it cannot safely switch the data stream to the new node, it keeps the connection to the previous node open and continues to send the data stream to that node until it has been safely sent.

Load balancing, in conjunction with indexer acknowledgment, is of key importance in a clustered deployment because it helps ensure that you don't lose any data in case of node failure. If a forwarder does not receive indexer acknowledgment from the node it is sending data to, it resends the data to the next available node in the load-balanced group.

Forwarders using the indexer discovery feature always use load balancing to send data to the set of peer nodes. You can enable weighted load balancing, which means that the forwarder distributes data based on each peer's disk capacity. For example, a peer with a 400GB disk receives twice the data of a peer with a 200GB disk. See "Use weighted load balancing."

For further information on:

- load balancing with indexer discovery, see "Use indexer discovery to connect forwarders to peer nodes."
- load balancing without indexer discovery, see "Set up load balancing" in the *Forwarding Data* manual.
- how load balancing works with indexer acknowledgment, read "Protect against loss of in-flight data" in the *Forwarding Data* manual.