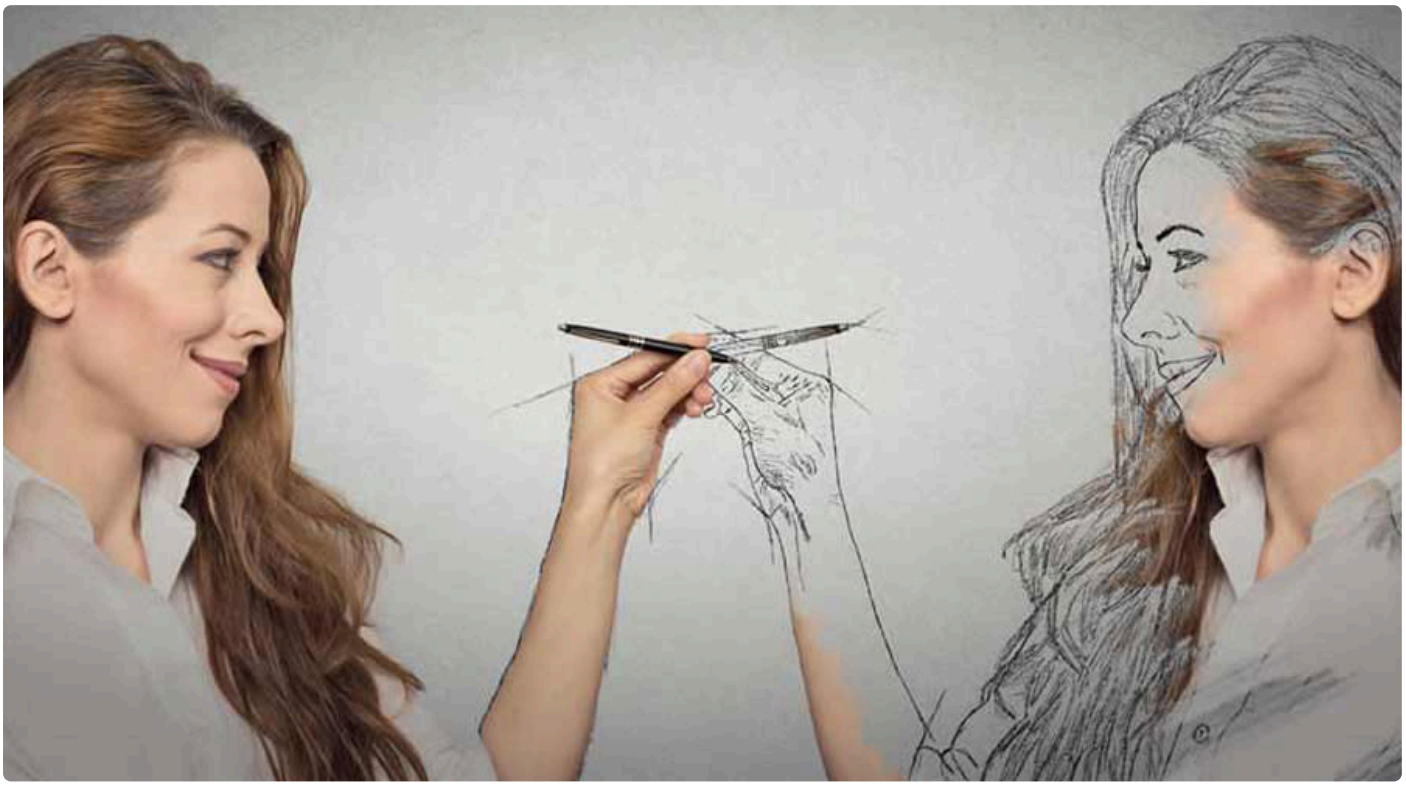
Managing the replica shards of the Search Guard index

Jochen Kressin · 2016-09-10



Search Guard stores nearly all it's configuration settings in a specially secured index directly in your Elasticsearch cluster.

This has the following advantages:

- You can hot-reload the complete settings without the need to restart any of your nodes
  - for example to add/remove users, or to change the permissions for a certain role
- You do not need to place any configuration files on your nodes
  - No need to keep the files in sync in case of changes
  - Simplified provisioning if you use Puppet, Ansible etc. to set up your environment

The Search Guard index is specially secured in the sense that you cannot simply read and write to it. Since it contains sensitive information, you need to provide a valid, known TLS certificate and use the sgadmin.sh script to update it.

However, apart from that, the Search Guard index is no different from any other index you store on your cluster. Which also means that you can run into sharding and replication issues, especially if you add and remove nodes dynamically from your cluster. Ending in the dreaded *Search Guard not initialized (SG11)* message.

> In almost all cases, the root cause is that neither a primary shard nor a replica shard of the Search Guard index is available.

If you are not familiar with the terms Shard and Replica, you might want to read about how Elasticsearch is storing data, and what you need to know about scaling. There's a good article in the Elasticsearch guide about these topics.

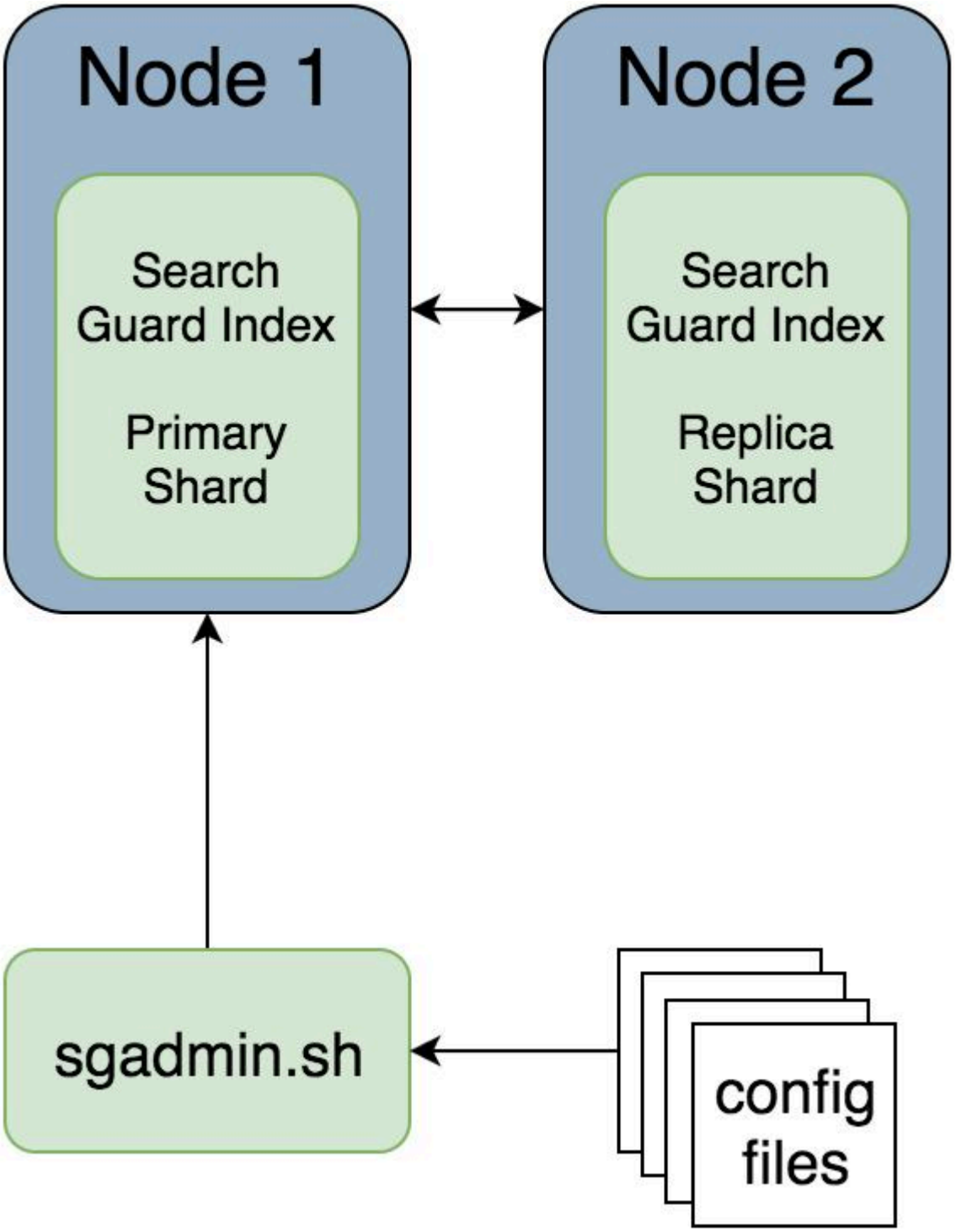
Let's explain the details by an example.

Initial cluster with two nodes

In the initial state, our cluster consists of two nodes. Both nodes are up and running, but there is no Search Guard index yet. If you execute sgadmin **for the first time** and therefore push the configuration to either one of those nodes, the following happens:

- Search Guard verifies the TLS certificate for security reasons
- Since the Search Guard index does not exist yet, it's created and populated with the configuration settings
- Search Guard determines the number of nodes participating in the cluster (only if index did not exist)
- Search Guard sets the number of replica shards accordingly (only if index did not exist)

In our case, we would end up with one primary shard and one replica shard. One node holds the primary shard, the other the replica.

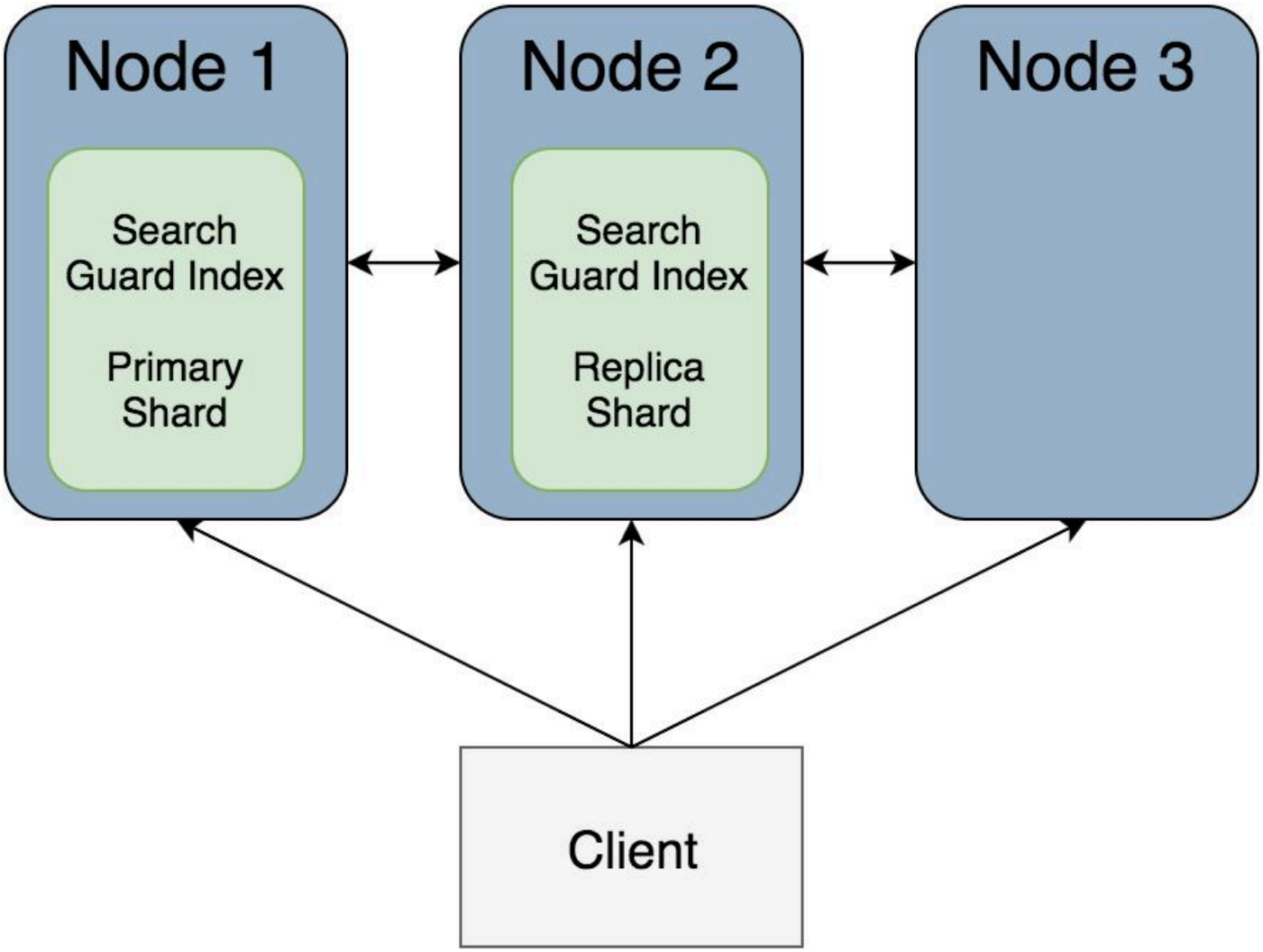


This is perfectly fine, and we can shut down any of the nodes and bring it up again, in any order. Since we have a shard on all of the participating nodes, the index will always be accessible.

Adding another node

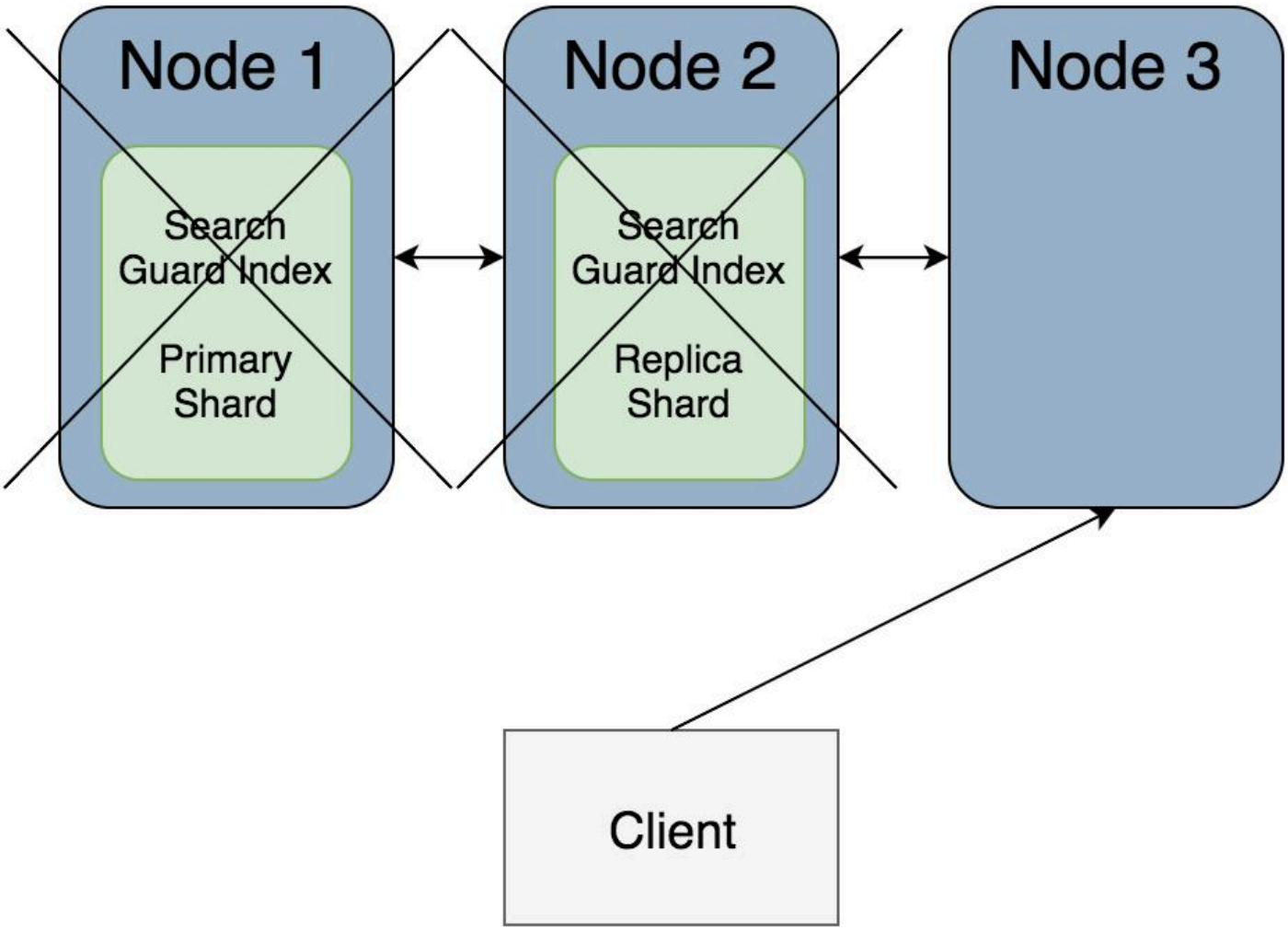
Now we add a third node to our cluster. If we do not do anything special, the node will join the cluster, but the *number*of*replicas* is still set to 1. What this means is that the third node will not receive a replica (copy) of the Search Guard index.

As long as a primary or replica shard is available on any other node, this is still perfectly fine. If you issue queries against the third node, the one without the replica, it will get the Search Guard index data from one of the two other nodes.



Removing nodes

Now imagine that for some reason you need to shut down the first two nodes. Now your cluster consists only of the third node that we added last. Since this node does not have a replica, and there are no other nodes left in the cluster with either primary or replica shard, Search Guard will complain about the missing index. And that means: *Search Guard not initialized (SG11).*



Two ways of handling replica shards

In order to mitigate this problem, you have two choices.

Updating the number of replicas manually

Whenever you add or remove nodes from your cluster, set the number of replica shards manually via sgadmin. You can use the –us parameter to set it. This can be done without changing the current configuration settings. For example, if you have 6 nodes in your cluster, and want to have a replica shard on all nodes, you'd set it to 5 like so:

```
/sgadmin.sh -ts truststore.jks -tspass password -ks keystore.jks -kspass password -icl -us 5
```

If you do not specify the configuration directory or individual config files, Search Guard will only update the number of replicas, but leave the current configuration settings alone.

Updating the number of replicas automatically

With the release of Search Guard 2.x.x.6, we will introduce a new, currently experimental feature called *auto*expand_. If you enable it, Search Guard will monitor changes in your cluster topology, and adjusts the number of replicas automatically whenever a node joins or leaves the cluster.

Search Guard will make sure that each node in the cluster will have a primary or replica shard of the Search Guard index. In most cases, this will eliminate the need to worry about shards not being available completely. If you have a huge number of nodes with a lot of redundancy in your cluster, you still might want to manage the number of replicas manually. That's why this feature will be configurable.

Update 18.09.2016: Search Guard 2.x.x.6 is released. You can use the switches -era (enable replica auto expand) and -dra (disable replica auto expand) to enable and disable this feature:

```
/sgadmin.sh -ts truststore.jks -tspass password -ks keystore.jks -kspass password -icl -era
```

Summary

If you run into the *Search Guard not initialized (SG11)* message after you have executed sgadmin, the most likely cause is that no shard for the Search Guard index is available. In most cases, this happens after you added or removed nodes dynamically. You can set the number of replica shards either by using the -us parameter of the sgadmin script. Or, beginning with 2.x.x.6, let Search Guard manage the number of replicas automatically by enabling the *auto*expand_ feature.

Image: shutterstock / pathdoc

This article was published on the Search Guard blog: https://search-guard.com/blog/search-guard-index-replica-shards/