

Cluster-Einrichtung

[Stand: 21.05.2019 | Programmversion: 12.00.000]

Der Betrieb eines Clusters kann Sinn ergeben, um Hochverfügbarkeit oder Lastverteilung zu erreichen. Dieses Dokument beschreibt, wie man ELOenterprise einrichtet, um ELOenterprise in einem Cluster zu betreiben.

Dabei beschränkt es sich auf die Konfiguration der ELO Module und beschreibt nicht die für den Clusterbetrieb notwendige Einrichtung von Drittsoftware wie Datenbanken, http-Proxies zur Lastverteilung oder Failover-Manager.

Inhalt

1	Einführung Cluster	2
2	ELO Server Setup	3
2.1	Konfigurationsdateien	3
2.2	Instanznamen	3
3	Einrichtung Cluster.....	4
3.1	Aktiv/Passiv HA-Cluster.....	4
3.2	Aktiv/Aktiv HA-/Loadbalancing-Cluster	4
4	ELO iSearch	6
4.1	Master-Nodes und Split-Brain	6
4.2	Getrennte Cluster im selben Netzwerk	6
4.3	Ausfallsicherheit und Leistung	7
4.3.1	Shards.....	7
4.3.2	Replicas	8
4.4	Failover-Cluster	9

1 Einführung Cluster

Ein Cluster zur Lastverteilung hat den Zweck die Arbeitslast eines Dienstes auf mehrere Rechner zu verteilen, um die Antwortzeiten zu reduzieren. Clients können dann statisch bestimmten Diensten zugeordnet werden, doch kann dies zu ungleicher Verteilung von Rechenaufgaben führen. Um dem entgegen zu wirken, gibt es Software zur Lastverteilung wie z.B. Apache httpd mit `mod_proxy_balancer`¹, welche Anfragen dynamisch auf die zur Verfügung stehenden Rechner verteilt.

Unter einem Hochverfügbarkeitscluster (High-Availability-Cluster – HA-Cluster) versteht man den Verbund von mindestens zwei Rechnern, die die Ausfallsicherheit eines Dienstes erhöhen sollen. Im Fehlerfall eines Rechners werden die darauf verfügbaren Dienste auf noch verfügbare Rechner migriert. Einen HA-Cluster kann auch mit einer Lastverteilung kombiniert werden, hierbei spricht man dann von einem aktiv/aktiv-Cluster. Steht dagegen im Hintergrund ein Standby-System bereit, welches erst dann einspringt, wenn das Primärsystem ausfällt, so wird dies als aktiv/passiv- oder Failover-Cluster bezeichnet.

¹ https://httpd.apache.org/docs/2.4/mod/mod_proxy_balancer.html

2 ELO Server Setup

Das ELO Server Setup unterstützt die Installation von ELOenterprise für Clustersysteme. Jedoch ist nicht jeder beliebige Konfigurationsfall abgedeckt. Im Folgenden werden ein paar Kniffe vorgestellt, wie man das ELO Server Setup verwenden kann, um gewünschte Clusterkonfigurationen zu realisieren.

2.1 Konfigurationsdateien

Das ELO Server Setup verwendet zwei Dateien zur Konfiguration, welche inhaltlich meist identisch, semantisch aber unterschiedlich sind. Die erste Datei wird unter `%HOMEPATH%/.elosetup.conf` (bzw. `${HOME}/.elosetup.conf`) abgelegt. Änderungen in der Web-Oberfläche werden beim Wechseln des Tabs im ELO Server Setup sofort in die erste Datei geschrieben. Sie repräsentiert den Soll-Stand der ELO Installation.

Die andere Datei wird im Verlauf kurz vor dem Abschluss der Installation nach `<ELO>/config/serversetup2/elosetup.conf` geschrieben und repräsentiert damit den Ist-Stand der ELO Installation.

Als aktuellere der beiden Dateien wird diejenige mit dem neueren Zeitstempel betrachtet. Das hat jedoch nur Auswirkungen darauf, was in der Web-Oberfläche angezeigt wird. Diese Eigenschaft ist nützlich, wenn es mehrere Administratoren gibt, die Änderungen über das Setup durchführen, denn das ELO Server Setup verteilt getätigte Änderungen an die jeweils anderen Administratoren, sobald diese es starten.

2.2 Instanznamen

Um für ein Archiv verschiedene Instanzen derselben Web-Anwendungen zu unterscheiden, werden sie in ihrer Konfigurationsdatei `config.xml` benannt. Dieser Mechanismus ist notwendig für den ELO Access Manager, den ELO Document Manager sowie den ELO Indexserver. Diese Web-Anwendungen haben in der Datenbank zusätzliche Konfigurationsparameter, welche je nach Instanz unterschiedlich sein müssen.

Ein Beispiel hierfür wäre der Parameter `privateUrlBase` für den ELO Indexserver. Dieser Parameter muss jeweils auf die URL gesetzt werden, über die genau diese Instanz des ELO Indexserver erreicht werden kann, damit sich andere Instanzen damit verbinden können, um die internen Zwischenspeicher von z. B. Masken zu synchronisieren.

Das ELO Server Setup richtet automatisch den Instanznamen für die Web-Anwendungen ein. Als Name wird der Name des ELO Application Servers gewählt, auf welchem sie ausgerollt wurden. Das ELO Server Setup stellt sicher, dass diese Namen eindeutig sind.

3 Einrichtung Cluster

3.1 Aktiv/Passiv HA-Cluster

Ein aktiv/passiv HA-Cluster benötigt verständlicherweise einen aktiven und einen passiven Part. Da der passive Part zur Laufzeit nur im Fehlerfall des aktiven Parts einspringt, ist es nicht notwendig, dass die beiden Teile voneinander wissen. Eher das Gegenteil ist der Fall, dass es sogar vorteilhaft ist, wenn die beiden Teile nichts voneinander wissen. Da davon ausgegangen werden kann, dass zur Laufzeit nur je eine Instanz erreichbar sein kann, würde der Versuch einer Synchronisation mit der nicht erreichbaren anderen Instanz immer fehlschlagen. Besser ist es, den Versuch erst gar nicht durchzuführen.

Damit die Instanzen jeweils unbekannt untereinander sind, sollen für einen aktiv/passiv HA-Cluster identische Instanznamen verwendet werden, was identische Installationen auf dem aktiven und passiven System voraussetzt. Weil das ELO Server Setup aber eindeutige Instanznamen (Namen der unterschiedliche ELO Application Server) voraussetzt, darf auch das ELO Server Setup nichts von der jeweils anderen Installation wissen. Erreicht werden kann dieser Zustand, indem die Installation auf dem aktiven System ohne Wissen des passiven Systems konfiguriert und installiert wird. Im Anschluss kopiert man die `${HOME}/.elosetup.conf` des aktiven Systems auf das passive System und führt dort die Installation unverändert ebenso aus.



Beachten Sie: Falls sich der Hostname des passiven Systems von dem des aktiven Systems unterscheidet und Sie den Namen im ELO Server Setup eintragen, gilt Folgendes:

Die Web-Oberfläche des ELO Server Setups passt die Namen der ELO Application Server an, sollten diese den vorigen Hostnamen beinhalten haben. Sie müssen die Namen der ELO Application Server dann auf den alten Wert zurücksetzen.

3.2 Aktiv/Aktiv HA-/Loadbalancing-Cluster

Im Gegensatz zum aktiv/passiv-Modus muss zum Beispiel der ELO Indexserver oder die ELO iSearch von den jeweils anderen Instanzen wissen und sich mit ihnen absprechen. Dazu konfigurieren Sie auf dem ersten Rechner im ELO Server Setup alle beteiligten Komponenten, auch die Komponenten auf den übrigen, am Cluster beteiligten Rechnern. Nach der Installation lassen sich die übrigen Rechner durch Kopieren der Datei `<ELO>/config/serversetup2/elosetup.conf` vom ersten Rechner auf die übrigen Rechner nach `%HOMEPATH%/.elosetup.conf` (bzw. `${HOME}/.elosetup.conf`) über das ELO Server Setup installieren.



Information: Die Konfigurationsdatei muss aus dem Installationsverzeichnis kopiert werden, da nach der ersten Installation nur diese Datei die vom ELO Server Setup generierten Zertifikate zur Verschlüsselung der Kommunikation zwischen ELO Indexserver und ELO iSearch beinhaltet.

4 ELO iSearch

Wenn Sie über das ELO Server Setup für ein Cluster mehrere ELO iSearches installiert haben, formen diese von sich aus automatisch ein eigenes Cluster. Für einfache Lastverteilungsszenarien sollten die vom ELO Server Setup angelegten Einstellungen ausreichend sein. Bei steigenden Anforderungen an die Ausfallsicherheit sollten, wie im Folgenden beschrieben, manche Parameter der Konfiguration angepasst werden. Die entsprechende Konfigurationsdatei finden Sie unter:

```
<ELO>/config/elastic/<service>/elasticsearch.yml.
```

4.1 Master-Nodes und Split-Brain

Um einen Cluster zu verwalten, bestimmen alle im ELO iSearch-Cluster beteiligten Instanzen (Nodes) einen Master-Node. Ein Node ist wählbar zu einem Master-Node, wenn in seiner Konfigurationsdatei der Parameter `node.master` den Wert `true` hat, was auch der Standardeinstellung entspricht, sollte der Parameter fehlen.

Kommt es zu einer Kommunikationsstörung in einem Cluster von zwei Nodes zwischen den beiden Instanzen, wobei beide Instanzen selbst noch lauffähig sind, so kommt es zu einer Split-Brain-Situation. Jeder der beiden Instanzen denkt, sie wäre der Master-Node und lässt schreibende Operationen auf ihrem Index zu. Dadurch divergieren jedoch die Daten der beiden Instanzen. Die Folge aus dieser Situation kann Datenverlust beim ersten Neustart einer Instanz sein, da sich diese dann den Datensatz der jeweils anderen Instanz in ihren lokalen Speicher holt. Dadurch gehen die während der Split-Brain-Situation getätigten Schreiboperationen verloren.

Eine solche Split-Brain-Situation lässt sich durch die Einstellung `discovery.zen.minimum_master_nodes: n` vermeiden. Als Parameter erwartet Sie eine Zahl n , die der abgerundeten Hälfte der wählbaren Master-Nodes plus eins entsprechen sollte (Beispiel für 5 wählbare Master-Nodes: $\lfloor \frac{5}{2} \rfloor + 1 = 3$). Die Einstellung sagt aus, dass der Cluster solange als intakt gilt, solange mindesten n wählbare Master-Nodes erreichbar sind. Ist ein Cluster nicht intakt, bekommt er den Zustand „rot“. Eine Instanz in einem solchen beschädigten Cluster erlaubt keine Operationen mehr. Würde der Cluster nun in zwei Teile aufbrechen, wäre der Teil weiter funktionsfähig, in welchem mehr als die Hälfte der Master-Nodes verblieben sind. Der kleinere Cluster wäre dann nicht mehr intakt. Die Standardeinstellung für diesen Parameter ist 1.

4.2 Getrennte Cluster im selben Netzwerk

ELO unterstützt die Installation getrennter Cluster im selben Netzwerk. Obwohl die Einstellung `cluster.name: ELO` gesetzt wird und bei allen Installationen identisch ist, können sich die Elasticsearch-Instanzen getrennter Installationen über den eingebauten Auto-Discovery-Modus nicht finden, da die Elasticsearch-Erweiterung Search Guard Verbindungen nur von Instanzen mit identischem Root-Zertifikat zulässt.

4.3 Ausfallsicherheit und Leistung

Die Elasticsearch bietet zwei wesentliche Konfigurationsparameter an, mit denen die Performance zum einen und die Ausfallsicherheit zum anderen verbessert werden können. Die beiden folgenden Abschnitte beschreiben, wie durch Erhöhen der Anzahl an Shards die Leistung und durch Erhöhen der Anzahl an Replicas zusätzlich auch die Ausfallsicherheit eines Clusters verbessert werden kann. Je nach Szenario bedingen beide Einstellungen einander.

4.3.1 Shards

Elasticsearch teilt Indices in eine konfigurierbare Anzahl an Teilen, sogenannte Shards, auf. Shards sind die kleinste Einheit an Daten, die im Cluster auf Knoten verteilt werden. Um Anfragen an einen Index auf mehrere Rechner zu verteilen, muss er mit mehr als einem Shard erzeugt worden sein oder mindestens eine Replica besitzen. Eine Änderung der Anzahl an Shards in einem Index erfordert eine Re-Indexierung.



Information: Jeder Index besteht aus (mindestens) einem Shard. Daraus folgt, dass, ohne manuellen Eingriff, bereits durch das Hinzunehmen einer zweiten Elasticsearch-Instanz in den Cluster, indexübergreifende Suchen auf die beiden Instanzen verteilt werden.

Bei ELO wird seit ELO 11 pro Maske ein Index angelegt. In der Standardeinstellung besteht jeder Index aus einem Shard. Durch die Aufteilung in einen Index pro Maske wird auch im Cluster schon ein Performance-Gewinn erreicht, da die Indices auf mehrere Nodes aufgeteilt werden.

Zusätzlich ergibt es Sinn, besonders große Indices in mehrere Shards aufzuteilen. Der Vorteil liegt darin, dass Suchanfragen dann parallel auf allen Shards ausgeführt werden. Allerdings werden maximal so viele Suchanfragen innerhalb eines Nodes parallel ausgeführt wie CPUs vorhanden sind.

Die Anzahl an Shards richtet sich demnach nach den enthaltenen Daten. Elasticsearch empfiehlt eine Shard-Größe von 20GB – 40GB.² Zu kleine Shards führen zu einem Overhead an Kommunikation zwischen den Shards, da die Daten zwischen den Shards synchronisiert werden müssen. Masken, die sehr viele SORDs mit Volltext enthalten, sollten demnach möglichst in mehrere Shards aufgeteilt werden.

Die Anzahl an Shards pro Maske kann über die Konfigurationsseite der ELO iSearch vorgenommen werden, erfordert allerdings eine Re-Indexierung aller Daten.

² <https://www.elastic.co/blog/how-many-shards-should-i-have-in-my-elasticsearch-cluster>

4.3.2 Replicas

Replicas sind sozusagen Kopien eines Index. Es kann jeder beliebige Wert gewählt werden, auch 0. Für jeden Shard wird ein Primary-Shard sowieso die angegebene Anzahl an Replica-Shards im Cluster erzeugt. Primary-Shards und Replica-Shards werden immer auf unterschiedlichen Nodes angelegt, um im Falle eines Ausfalls auch weiterhin die Daten bereitstellen zu können. Fällt ein Node aus, werden die Primary-Shards, die bisher auf diesem Node allokiert waren, auf einem anderen Node allokiert. Die Replica-Shards werden auf andere Nodes verteilt, um weiterhin die bestmögliche Ausfallsicherheit und Performance zu gewährleisten. Als Orientierung kann man davon ausgehen, dass die Anzahl an Replicas der Anzahl an Nodes entspricht, die ohne negative Auswirkungen ausfallen können.

Der Weitere Vorteil von Replica-Shards ist die Parallelisierung von Suchanfragen: Elasticsearch verteilt Suchanfragen auf unterschiedliche Primary-Shards und Replica-Shards, um die Last der Suchanfragen zu verteilen.

Der Health-Zustand eines Indices ist abhängig davon, welche Shards verfügbar sind: Ist der Primary-Shard vorhanden, ist der Zustand „gelb“. Sind zusätzlich alle Replica-Shards verfügbar, ist der Zustand „grün“.

Die Menge der Replica-Shards sollte an die Anzahl der Nodes im Cluster angepasst werden, sodass sich die Shards sinnvoll auf mehrere Nodes verteilen können, um Ausfallsicherheit zu gewährleisten. Andererseits benötigt jede Replica genauso viel Speicherplatz wie die Primary-Shards.

Die Standardeinstellung für die Anzahl an Replicas ist 1. So wird in einem Cluster von mehr als einer Instanz jeder Shard einmal repliziert, ist also insgesamt zweimal vorhanden. Eine Änderung an der Anzahl der Replicas erfordert keine Re-Indexierung.

Um eine Re-Indexierung in einem Cluster zu beschleunigen kann für die Dauer der Re-Indexierung die Anzahl an Replicas auf 0 gesetzt werden und nach dem Beenden erhöht werden. Ist die Anzahl an Replicas > 0 werden die Daten sowohl im Primary-Shard als auch in jedem Replica-Shard zeitgleich indexiert. Das bedeutet, dass die Analyse und Umwandlung in Elasticsearch-Daten mehrmals parallel stattfinden. Wird die Anzahl an Replicas auf 0 gesetzt, werden die Daten während der Re-Indexierung nur in einem Node verarbeitet. Sobald die Anzahl an Replicas erhöht wird, werden die schon indexierten Primary-Shards auf die anderen Nodes kopiert, was in der Regel deutlich schneller geht. Natürlich ist dadurch während der Re-Indexierung keine Ausfallsicherheit gewährleistet.

Die Einstellung ist bezogen auf einen Index und kann exemplarisch wie folgt für die Maske mit der ID 42 des Archivs *archiv* auf 2 geändert werden:

```
curl -X PUT "localhost:9200/archiv/42/_settings" -H 'Content-Type: application/json' -d '{
  "index" : {
    "number_of_replicas" : 2
  }
}'
```

Möchten Sie die Einstellung für alle Indices ändern, so können Sie die Angabe des Index weglassen und die URL `localhost:9200/_settings` verwenden.

4.4 Failover-Cluster

In einem Failover-Cluster, fehlen verständlicherweise der ELO iSearch im Stand-By-System aktuelle Daten aus dem laufenden Betrieb. Damit würde, im Failover-Fall, die ELO iSearch veraltete oder überhaupt keine Daten finden. Das Speichern des ELO iSearch-Index auf einem hochverfügbaren Netzlaufwerk ist daher unerlässlich. Im ELO Server Setup kann der Pfad gewählt werden, wo die ELO iSearch ihren Index speichern soll. Die Einstellung finden Sie im ELO Server Setup unter *Advanced Configuration > Application Servers* und bei dem entsprechenden Eintrag für die ELO iSearch *Data folder*.