ELO Analytics Sicherungsmanagement

[Stand: 16.05.2019 | Programmversion: 12]

Dieses Dokument beschreibt die Aspekte der Sicherung und Wiederherstellung von Benutzereinstellungen, Suchen, Visualisierungen und Dashboards von ELO Analytics.

Information: Dieses Dokument bezieht sich auf die folgenden Versionen:

- ELO Analytics: 12
- Kibana: 5.6.3

Dieses Dokument ist nur für die ELO Business Partner gedacht und stellt keine Endkundendokumentation dar.

Sie finden weitere Artikel zu ELO Analytics in der ELO Community. Schauen Sie dort vorbei und abonnieren Sie den ELO Analytics-Bereich, um keine Neuigkeit zu verpassen.

https://community.elo.com

Information: Mit ELO Analytics erhalten Sie weitere Dokumentationen aus dem ELO PartnerPortal.

- Technische Dokumentation
- Benutzerdokumentation
- Häufig gestellte Fragen
- Konfiguration
- Sicherungsmanagement (liegt Ihnen vor)
- Komponentendiagramm

Inhalt

1 Vorrausetzungen ... 3
2 Sicherungsmanagement ... 4
2.1 Snapshots in Elasticsearch ... 4
2.1.1 Konfiguration/Vorbereitung ... 4
2.1.2 Durchführung mithilfe von ELO Analytics ‚Dev Tools' ... 5
2.1.3 Snapshot-Konfiguration erstellen ... 6
2.1.4 Sicherung/Snapshots erstellen ... 7
2.1.5 Sicherung/Snapshots wiederherstellen ... 9
2.2 Automatische Sicherung mit Elasticsearch-Snapshots ... 10
2.2.1 Elasticsearch-Befehle mit cURL in der Kommandozeile ausführen ... 10
2.2.2 Die Skripte ‚backup.cmd' und ‚restore.cmd' ... 12
3 Weiterführende Literatur ... 22

1 Vorrausetzungen

Die hier gezeigten Lösungsansätze erheben keinen Anspruch auf eine vollständige Sicherungsstrategie für alle Eventualitäten. Sie sollen vielmehr einen schnellen Einstieg in die Sicherung und Wiederherstellung von ELO Analytics-Objekten bieten. Außerdem sollen die gezeigten Lösungsansätze helfen, die ELO Analytics-Daten in Ihr bestehendes Sicherungskonzept einzubinden.

Für die Sicherung sind folgende Kenntnisse erforderlich bzw. sollten erworben werden:

- Grundlagen im Umgang und Konfiguration von ELO allgemein, ELO Analytics und Elasticsearch.
- Umgang mit Elasticsearch-Abfragen in der Entwicklerkonsole von ELO Analytics.
- Batch Programmierung (unter Microsoft Windows) und Umgang mit dem Tool cURL zur Anpassung der Sicherungsskripte.

2 Sicherungsmanagement

Ab ELO Analytics 11 muss der ELO Analytics-Index in Elasticsearch gesichert werden. Der ELO Analytics-Index speichert alle von Ihnen erstellten ELO Analytics-Objekte (Musterindizes, Suchen, Visualisierungen und Dashboards). Sie haben jedoch im Rahmen Ihrer etablierten Sicherungsstrategie für ELO mehrere Möglichkeiten die ELO Analytics-Objekte zu sichern:

1. Sie können den gesamten Elasticsearch-Index im Dateisystem sichern (erfordert, dass der Elasticsearch-Dienst zuvor beendet wird).
2. Sie können einzelne Indizes im Dateisystem von Elasticsearch sichern (erfordert, dass der Elasticsearch-Dienst zuvor beendet wird).
3. Sie können die ELO Analytics-Objekte über den Tab *Management* in ELO Analytics exportieren und importieren.
4. Sie können den Snapshot-Mechanismus von Elasticsearch verwenden.

Im Rahmen dieser Dokumentation beschäftigen wir uns mit dem vierten Punkt: Der Nutzung des Snapshot-Mechanismus von Elasticsearch.

2.1 Snapshots in Elasticsearch

Snapshots erlauben eine Sicherung und Wiederherstellung eines oder mehrerer Elasticsearch-Indizes auf einem lokalen oder externen Dateisystem.

2.1.1 Konfiguration/Vorbereitung

Um Snapshots herstellen zu können, muss Elasticsearch konfiguriert werden. Im Folgenden speichern wir den Snapshot im lokalen Dateisystem, es ist jedoch genauso möglich externe Dateisysteme zu verwenden. Weitere Informationen finden Sie dazu in der Elasticsearch-Dokumentation.

2.1.1.1 Snapshots Dateisystem-Pfad in Elasticsearch freigeben

Bearbeiten Sie dazu die Elasticsearch-Konfigurationsdatei *elasticsearch.yml*.

Um einen Dateisystempfad verwenden zu können, müssen Sie Elasticsearch erst erlauben, diesen zu verwenden. Fügen Sie dafür einen oder mehrere Sicherungszielpfade ein. Es ist genauso möglich UNC-Pfade zu verwenden. Auch bei Windows-Pfaden müssen doppelte Backslashes angegeben werden (Escaping).

```
path.repo: ["C:\\ES Backup\\Snapshot"]
```

2.1.1.2 Search-Guard-Zugriff für Snapshot-Restore erlauben

Weiterhin müssen Sie dem Search-Guard-Plugin das Wiederherstellen von Backups explizit erlauben.

```
searchguard.enable_snapshot_restore_privilege: true
searchguard.check_snapshot_restore_write_privileges: false
```

2.1.2 Durchführung mithilfe von ELO Analytics ‚Dev Tools'

Im Folgenden wird das Entwicklerwerkzeug (*Dev Tools*) in ELO Analytics verwendet, um Befehle an die Elasticsearch zu senden. Diese werden nur angezeigt, wenn Sie mit einem ELO Hauptadministrator angemeldet sind.



Abb. 1: ELO Analytics ‚Dev Tools'

Um die Funktionsweise der Konsole kennenzulernen, führen Sie zum Einstieg folgende Befehle aus:

```
GET _cat/health?v=true
GET _cat/aliases?format=yaml
GET _cat/indices?v=true
```

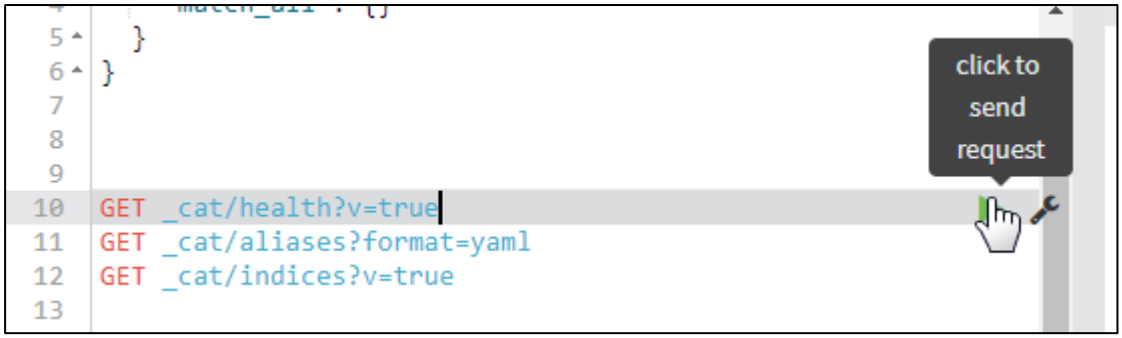


Abb. 2: Grüner Pfeil in der Konsole

Um einen einzelnen Befehl pro Zeile auszuführen, klicken Sie entweder auf den grünen Pfeil neben der ersten Zeile oder drücken STRG + ENTER.

Der Befehl *GET _cat/health?v=true* zeigt Ihnen den gesamten Zustand der Elasticsearch an. Der Parameter *v* gibt an, dass die Tabelle mit Überschriften angezeigt werden soll.

Der Befehl *GET _cat/aliases?format=yaml* gibt Ihnen alle Aliasnamen der Indizes aus (im Format YAML (Punktnotation) oder JSON). Lassen Sie den Parameter weg, wird eine Tabellendarstellung genutzt.

Der Befehl *GET _cat/indices?v=true* zeigt alle Elasticsearch-Indizes Ihrer ELO Installation an.

Es gibt neben *GET* noch weitere Befehle, die im Gegensatz zu *GET* Schreibzugriff in der Elasticsearch ausführen. Dazu gehören:

- PUT bzw. POST
    - Im Folgenden nutzen wir nur POST-Aufrufe, da wir nur neue Objekte erstellen (POST) und keine vorhandenen Objekte verändern wollen (PUT). Die Parameter für den Aufruf werden als JSON-Objekt an die POST-Zeile gefügt. Es ist dabei unerheblich, ob das JSON-Objekt in einer Zeile steht oder über mehrere Zeilen geht.

```
POST url
{ <JSON-Objekt>
}
```

- DELETE
    - Damit löschen Sie Objekte in der Elasticsearch ohne Nachfrage.

2.1.3 Snapshot-Konfiguration erstellen

Ein Snapshot muss zuerst konfiguriert und benannt werden (z. B. *my_backup*), bevor er ausgeführt werden kann. Dazu zählt das Zielsystem sowie den freigegebenen Ordnernamen, den wir zuvor Elasticsearch für Sicherungen in *elasticsearch.yml* erlaubt haben zu nutzen.

Dazu geben Sie folgenden Befehl ein und beachten dabei auch die doppelten Backslashes, welche in JSON notwendig sind:

```
POST /_snapshot/my_backup
{
  "type": "fs",
  "settings": {
     "compress": true,
     "location": "C:\\BACKUP\\Analytics"
  }
}
```

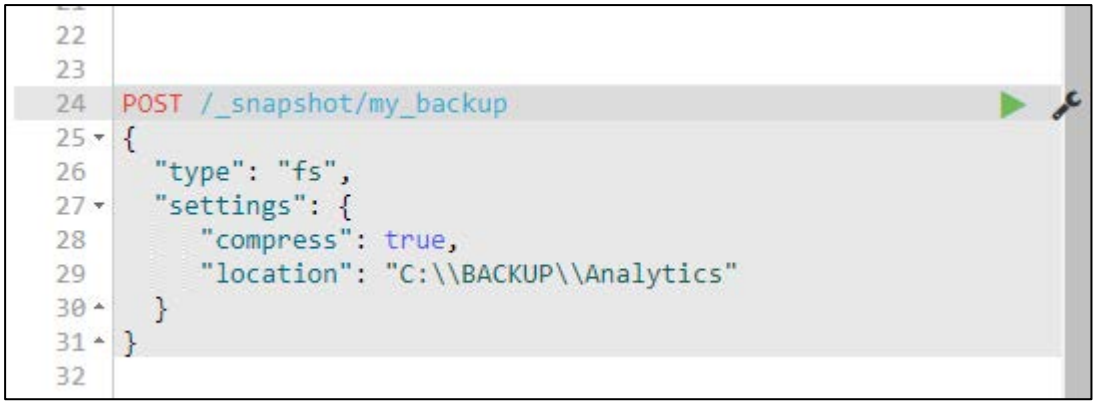


Abb. 3: POST-Befehl in der Konsole

Um den Befehl auszuführen, klicken Sie entweder auf den grünen Pfeil neben der ersten Zeile oder drücken Sie STRG + ENTER.

> **Information**: Wir empfehlen, dass Sie die schreibenden Befehle in einer eigenständigen ELO Installation testen.

Wenn Sie Elasticsearch korrekt konfiguriert haben, sehen Sie die folgende Meldung:

```
{
  "acknowledged": true
}
```

Sie können nun eine Sicherung durchführen.

Eine solche Snapshot-Konfiguration können Sie immer wieder überschreiben und ändern, ohne dass dies darin vorhandene Sicherungen ändert, sofern am Speicherort die notwendige Sicherung vorhanden ist. Um die aktuelle Konfiguration festzustellen, führen Sie einen GET-Befehl aus:

```
GET /_snapshot/my_backup
```

Weiterhin können Sie alle Snapshot-Konfigurationen mit *_all* erfahren:

```
GET /_snapshot/_all
```

> **Information**: Wenn Sie die Elasticsearch nach einem Datenverlust wiederherstellen, müssen Sie zuerst die Snapshot-Konfiguration ausführen, bevor Sie Ihre Sicherung wiederherstellen können.

2.1.4 Sicherung/Snapshots erstellen

Um eine Sicherung der ELO Analytics-Objekte zu erhalten, benötigen Sie Ihren Archivnamen. Jedes Archiv besitzt einen eigenen Elasticsearch-Index, welcher in der Form „.kibana_<archivname>“ in Kleinbuchstaben abgelegt ist (z. B. *demo1*).

Sie können, wie beim Indexmuster in ELO Analytics, den Sternoperator verwenden, um mehrere Indizes zu sichern (z. B. .kibana_*).

Als zweite Alternative können Sie auch einen Array mit definierten Indizes verwenden (z. B. [“index1“, “index2“]). Wenn Sie so mehrere Indizes sichern möchten, können Sie mit „ignore_unavailable“ vermeiden, dass ein Fehler geworfen wird, wenn ein oder mehrere Indizes nicht existieren.

Zusätzlich können Sie mit "wait_for_completion" auf das Ende des Backups warten. Beachten Sie jedoch, dass Sie trotzdem eine kurze Zeit warten müssen, um im Dateisystem auf alle Sicherungsdateien zugreifen zu können. Der Aufruf kehrt für automatische Skripte immer etwas zu früh zurück.

Beispiel:

```
PUT /_snapshot/my_backup/snapshot_1
{
  "indices": ".kibana_demo1",
  "ignore_unavailable": true,
  "include_global_state": false,
  "wait_for_completion" : false
}
```



Information: Mit Snapshots können Sie nicht den Index „Searchguard“ sichern, da dieser durch das Plugin *Search Guard* geschützt ist. Weiterhin können Sie den globalen Status (*include_globale_state*) zwar sichern, aber nicht wiederherstellen, da *Search Guard* dies grundlegend verbietet. Die Meldung lautet dazu:

„no permissions for cluster:admin/snapshot/restore“.

Einen solchen erstellten Snapshot können Sie nicht überschreiben (nur löschen), daher benötigen Sie immer einen neuen Namen. Um weitere Informationen zum Snapshot zu erhalten (wie z. B. gesicherte Indizes), führen Sie einen GET-Befehl aus. Sie finden darin auch den aktuellen Fortschritt der Sicherung im Schlüssel *state*. Dieser Schlüssel kann die Werte *IN_PROGRESS*, *SUCCESS*, *FAILED*, *PARTIAL* und *INCOMPATIBLE* enthalten. Weitere Informationen dazu finden Sie in der Elasticsearch-Dokumentation zu Snapshots.

```
GET /_snapshot/my_backup/snapshot_1
```

Weiterhin können Sie alle erstellten Snapshots auflisten, indem Sie *_all* verwenden.

```
GET /_snapshot/my_backup/_all
```

Da Sie Snapshots nicht überschreiben können, können Sie diese löschen. Damit werden auch alle Dateien gelöscht. Der Befehl lautet:

```
DELETE /_snapshot/my_backup/snapshot_1
```

Folgende Fehlermeldungen können auftreten:

Fehler:
"type": "cluster_block_exception", **"reason": "blocked by: [FORBIDDEN/4/index closed];"**
Lösung:

Der Index wurde geschlossen. Öffnen Sie den Index durch

```
POST /.kibana_<archivname>/_open
```

Fehler:

```
"type": "invalid_snapshot_name_exception",
"reason": "[analytics_backup:snapshot_6]Invalid snapshot name
[snapshot_6], snapshot with the same name already exists"      }
```

Lösung:

Ein Snapshot mit diesem Namen existiert bereits. Sie haben zwei Möglichkeiten:

a) Löschen Sie den Snapshot mit

```
DELETE /_snapshot/my_backup/snapshot_1
```

b) Ändern Sie den Namen des neuen Snapshots und wiederholen den Befehl.

2.1.5 Sicherung/Snapshots wiederherstellen

Sie können eine Sicherung wiederherstellen. Dazu müssen Sie jedoch zuerst einmal den Index schließen. Nach erfolgter Sicherung wird der Index wieder automatisch geöffnet.

Für den Kibana-Index des Archivs *Demo1* lautet damit der Befehl

```
POST /.kibana_demo1/_close
```

Die Wiederherstellung erfolgt dann mit dem nachfolgenden Befehl. Bereits existierende Indizes werden überschrieben.

```
POST /_snapshot/my_backup /snapshot_1/_restore{
  "indices": [".kibana_demo1"],
  "partial" : false,
  "include_aliases" : true
}
```

Sie können im Schlüssel *indices* wieder mehrere Indizes einsetzen oder den Sternoperator verwenden.

Mit dem Befehl *„partial" : false* wird angegeben, dass die Operation fehlschlägt, wenn die Wiederherstellung auf einem Shard missglückt.

Der Befehl *„include_aliases" : true* definiert, dass Index-Aliasse hergestellt werden sollen, falls diese existieren.

Zuletzt können Sie prüfen, ob die Wiederherstellung funktioniert hat, indem Sie den Status abfragen:

```
GET /_snapshot/my_backup/snapshot_1
```

Den neuen Index finden Sie dann auch in der Liste der Indizes:

```
GET _cat/indices/.kibana_demo1?v=true
```

2.2 Automatische Sicherung mit Elasticsearch-Snapshots

Die im vorigen Kapitel gezeigten Befehle können auch automatisch ausgeführt werden. Im folgenden Beispiel wird gezeigt, wie Sie die Elasticsearch-Befehle in der Kommandozeile (hier unter Windows) ausführen können.

2.2.1 Elasticsearch-Befehle mit cURL in der Kommandozeile ausführen

Bei den Elasticsearch-Befehlen handelt es sich um REST-Befehle, die über das Hypertext Transfer Protokoll (HTTP) gesendet werden. Da dies aus der Kommandozeile nicht möglich ist, verwenden wir dazu das oft eingesetzte Programm cURL (Client for URLs). Für Windows müssen Sie es zuerst herunterladen (oder die im Beispielskript beigelegte Version verwenden) und in den Ordner Ihres Skriptes kopieren.

Um einen REST-Aufruf mit cURL durchzuführen, geben Sie die nachfolgende Zeile in die Kommandozeile ein. Dazu benötigen Sie zuvor noch Ihren Servernamen und Port der Elasticsearch sowie das Elasticsearch-Kennwort, welches Sie im ELO Server Setup angegeben haben (im Beispiel „elo"). Beachten Sie auch, dass Sie auf die Elasticsearch, wie empfohlen, ggfs. nicht von außen zugreifen können.

```
curl http://admin:elo@servername:9200/_cat/indices?v=true
```

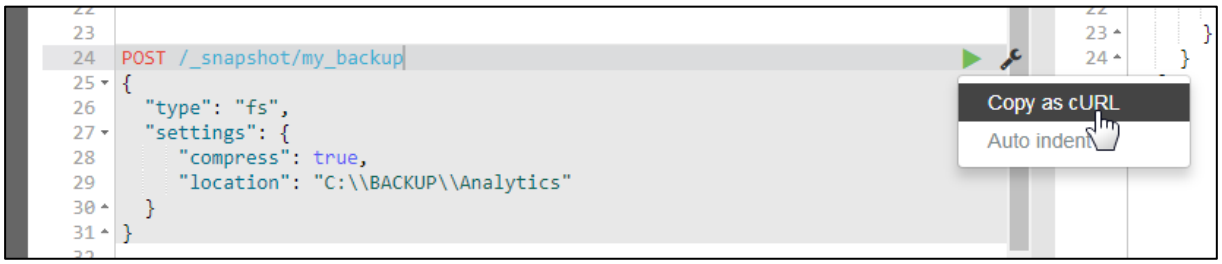


Abb. 4: CURL-Befehl in der Konsole erzeugen

Um eine Snapshot-Konfiguration zu erstellen, müssen wir jedoch einen POST-Befehl abschicken. In den ELO Analytics-Dev-Tools können Sie sich zu einem Befehl auch den CURL-Befehl generieren lassen. Klicken Sie dazu auf das Schraubenschlüsselsymbol und wählen *Copy as cURL* aus.

Allerdings erzeugt dieser Befehl für die Windows-Kommandozeile eine ungültige Syntax und muss angepasst werden:

```
curl -XPOST "http://servername:9200/_snapshot/my_backup" -H 'Content-Type:
application/json' -d'
{
  "type": "fs",
  "settings": {
     "compress": true,
     "location": "C:\\BACKUP\\Analytics"
  }
}'
```

Um die Syntax für Windows anzupassen, sind folgende Änderungen notwendig:

1. Benutzername und Kennwort vor dem Servernamen hinzufügen.
2. Hochkommata durch doppelte Hochkommata ersetzen.
3. Doppelte Hochkommata in JSON durch Backslash maskieren.

Nach den Änderungen sieht der Befehl dann folgendermaßen aus (Anpassungen in Rot):

```
curl -XPOST "http://admin:elo@servername.local:9200/_snapshot/my_backup" -H
"Content-Type: application/json" -d"
{
  \"type\": \"fs\",
  \"settings\": {
     \"compress\": true,
     \"location\": \"C:\\BACKUP\\Analytics\"
  }
}"
```

Sie können jedoch einen Befehl mit mehreren Zeilen nicht ohne Weiteres in der Kommandozeile eingeben. Sie müssen dazu alles in eine Zeile schreiben:

```
curl -XPOST "http://admin:elo@servername.local:9200/_snapshot/my_backup" -H
"Content-Type: application/json" -d"{  \"type\": \"fs\",  \"settings\": {
\"compress\": true,     \"location\": \"C:\\BACKUP\\Analytics\"  }}"
```

Alle weiteren Elasticsearch-Befehle können genauso durchgeführt werden. Sie können auch JSON-Formatierer im Internet verwenden, um JSON in einer Zeile zu formatieren.

2.2.2 Die Skripte ‚backup.cmd' und ‚restore.cmd'

Die Windows-Skripte *backup.cmd* und *restore.cmd* ermöglichen Ihnen eine einfache Sicherung und Wiederherstellung eines einzelnen Kibana-Indexes. Sie sind sehr einfach gehalten und lassen sich nur direkt und ohne Parameter konfigurieren. Sie können die Skripte an Ihre eigenen Erfordernisse und Sicherungsstrategien anpassen, verändern oder Ideen für eigene Skripte extrahieren.

Die Skripte nutzen das aus dem vorherigen Kapitel bekannte cURL sowie 7-ZIP, um die Sicherungsdateien in eine einzelne Datei zu sichern und den Sicherungspfad für die nächste Sicherung freizumachen (damit muss der Snapshot-Name nicht jedes Mal geändert werden).

Um jedoch mehrere Archive zu sichern, machen Sie sich Kopien von den Dateien und passen Sie die Kopien separat an.

Die Konfiguration finden Sie am Anfang der Skripte. Jeder Konfigurationsschlüssel ist kommentiert. Im Folgenden stellen wir Ihnen eine Beschreibung der Einstellungen vor.

Schlüsselname	Beschreibung	Beispiel
ES_USERNAME	Benutzername für Elasticsearch (admin)	admin
ES_PASSWORD	Kennwort für Elasticsearch-Zugriff im ELO Server Setup definiert	
PROTOCOL	http oder https	
ES_SERVER	Der Elasticsearch-Servername	
ES_PORT	Port von Elasticsearch	9200
ES_SNAPSHOT_CONFIG_NAME	Konfigurationsname für Snapshots	automatic_snapshot_1
ES_SNAPSHOT_NAME	Sicherungsname des Snapshots. Dieser Name bleibt immer gleich, da die Sicherungsdateien in ein ZIP-Archiv gesichert und danach gelöscht werden.	
ELO_ARCHIVE_NAME	Ihr ELO Archivname in KLEINBUCHSTABEN!	demo1 repository

ES_SNAPSHOT_LOCATION	Absoluter Pfad wie er in der Datei *elasticsearch.yml* eingetragen ist.	
SNAPSHOT_REMOTE_LOCATION	Absoluter Pfad, der entweder wie der Pfad ES_SNAPSHOT_LOCATION lauten kann, wenn das Skript auf dem Elasticsearch-Server läuft, oder der eine Verzeichnisfreigabe sein kann (Benutzerzugriff muss schreibend sein).	\\share\name Für den Skriptbenutzer können Sie mit *net use* einen Zugriff anlegen.
ZIP_LOCATION	Speicherort und Dateiname des zu erstellenden Skriptes	\\Backupserver\analytics.7z)
ZIP_APP	ZIP-Kommandozeilen-programm mit Parameter. Ggfs. müssen Sie das Skript direkt anpassen, wenn Sie ein anderes Sicherungsprogramm einsetzen möchten.	

2.2.2.1 Umsetzung von ‚backup.cmd'

Das Skript *backup.cmd* prüft zuerst, ob die Elasticsearch-Verbindung funktioniert, erstellt dann eine Snapshot-Konfiguration und sichert darin einen einzelnen Kibana-Index. Alle Sicherungsdateien werden dann in ein ZIP-Archiv **verschoben**, sodass ein weiterer Aufruf von *backup.cmd* nicht fehlschlägt.

2.2.2.2 Beispielskript ‚backup.cmd'

```
@ECHO OFF
REM This script is an example how to backup the ELO Analytics/kibana index
of a specific Analytics archive.
REM You may/should/must adapt it to your needs, environment and backup
strategy.
REM You need a copy of this file for another archive if you have multiple
archives.
REM
REM Be aware that this script is just plain simple and may not work
properly under certain instances!
REM
REM You may use, copy and change it according to your needs on your own
risk.
REM


REM Extract the Applications: src\curl.exe
REM Curl Download:
https://dl.uxnr.de/build/curl/curl_winssl_msys2_mingw64_stc/curl-
7.59.0/curl-7.59.0.zip
REM 7ZIP Download: http://www.7-zip.de/download.html

REM Define your server connection here
SET ES_USERNAME=admin
SET ES_PASSWORD=elo
SET PROTOCOL=http
SET ES_SERVER=eloanalytics.local
SET ES_PORT=9200

REM Name the ES snapshot configuration (any name)
SET ES_SNAPSHOT_CONFIG_NAME=analytics_backup
REM This snapshot (backup of kibana index) will be named :
SET ES_SNAPSHOT_NAME=automatic_snapshot_1
REM Set archive name here which defines the name of the kibana index to be
backed up
SET ELO_ARCHIVE_NAME=demo1

REM Where shall ES store its backup. This path must be allowed in
elasticsearch.yml
REM elasticsearch.yml
REM path.repo: ["C:\\BACKUP\\Analytics"]
REM
REM searchguard.enable_snapshot_restore_privilege: true
REM searchguard.check_snapshot_restore_write_privileges: false
SET ES_SNAPSHOT_LOCATION=C:\\BACKUP\\Analytics
REM The script can be run on another computer, so we need the location of
ES_SNAPSHOT_LOCATION on this computer if shared
REM But if the script is run on the same computer just this:
REM SNAPSHOT_REMOTE_LOCATION=%ES_SNAPSHOT_LOCATION%
SET SNAPSHOT_REMOTE_LOCATION=\\eloanalytics.local\Analytics\*
```

```
REM ZIP Backup name and location
SET ZIP_LOCATION=backup.7z
REM 7ZIP command line arguments
SET ZIP_APP=7z a -r -sdel


REM ... HERE WE GO ...


SET FIRST_PART=curl
SET
ES_SERVER_PART=%PROTOCOL%://%ES_USERNAME%:%ES_PASSWORD%@%ES_SERVER%:%ES_POR
T%

SETLOCAL

echo.
echo Checking ES health
%FIRST_PART% -f %ES_SERVER_PART%%/_cat/health
IF ERRORLEVEL 1 (
  echo Error while calling:
  GOTO EXIT
)
ENDLOCAL

SETLOCAL

echo.
echo Creating Snapshot Configuration
REM POST /_snapshot/analytics_backup
REM {
  REM "type": "fs",
  REM "settings": {
     REM "compress": true,
      REM "location": "C:\\BACKUP\\Analytics"
  REM }
REM }
SET OPTIONS=-H "Content-Type: application/json" -d "{ \"type\": \"fs\",
\"settings\": { \"compress\": true, \"location\":
\"%ES_SNAPSHOT_LOCATION%\" }}"
%FIRST_PART% -XPOST %ES_SERVER_PART%/_snapshot/%ES_SNAPSHOT_CONFIG_NAME%
%OPTIONS% -f
IF ERRORLEVEL 1 (
  echo Error while calling:
  GOTO EXIT
)

echo.
echo.
echo Closing and Reopening Kibana Index
```

```
REM POST /.kibana_demo1/_close
%FIRST_PART% -XPOST %ES_SERVER_PART%/.kibana_%ELO_ARCHIVE_NAME%/_close -f
IF ERRORLEVEL 1 (
  echo Error while calling:
  echo %FIRST_PART% -POST
%ES_SERVER_PART%/.kibana_%ELO_ARCHIVE_NAME%/_close -f

  GOTO EXIT
)
REM POST /.kibana_demo1/_open
%FIRST_PART% -XPOST %ES_SERVER_PART%/.kibana_%ELO_ARCHIVE_NAME%/_open -f
IF ERRORLEVEL 1 (
  echo Error while calling:
  echo %FIRST_PART% -XPOST
%ES_SERVER_PART%/.kibana_%ELO_ARCHIVE_NAME%/_open -f
  GOTO EXIT
)
ENDLOCAL

echo.
echo Creating Backup
REM POST /_snapshot/analytics_backup/snapshot_1
REM {
  REM "indices": [".kibana_*"],
  REM "ignore_unavailable": false,
  REM "include_global_state": false,
  REM "wait_for_completion" : true
REM }
SET OPTIONS=-H "Content-Type: application/json" -d
"{\"indices\":[\".kibana_%ELO_ARCHIVE_NAME%\"],\"ignore_unavailable\":false
,\"include_global_state\":false,\"wait_for_completion\":true}"
%FIRST_PART% -XPOST
%ES_SERVER_PART%/_snapshot/%ES_SNAPSHOT_CONFIG_NAME%/%ES_SNAPSHOT_NAME%?wai
t_for_completion=true %OPTIONS% -f
IF ERRORLEVEL 1 (
  echo Error while calling:
  echo %FIRST_PART% -XPUT
%ES_SERVER_PART%/_snapshot/%ES_SNAPSHOT_CONFIG_NAME%/%ES_SNAPSHOT_NAME%
%OPTIONS% -f
  GOTO EXIT
)

echo.
echo Waiting 10s for Completition
ping 127.0.0.1 -n 11 > nul

echo. Zip whole snapshot folder to file %ZIP_LOCATION%
%ZIP_APP% %ZIP_LOCATION% %SNAPSHOT_REMOTE_LOCATION%
IF ERRORLEVEL 1 (
  echo Error while calling:
  echo %ZIP_APP% %ZIP_LOCATION% %SNAPSHOT_REMOTE_LOCATION%
```

```
  GOTO EXIT
)

:EXIT
```

2.2.2.3 Umsetzung von ‚restore.cmd'

Das Skript *restore.cmd* prüft zuerst, ob die Elasticsearch-Verbindung funktioniert, **löscht** alle Dateien im Remote-Backup-Ordner (SNAPSHOT_REMOTE_LOCATION), entpackt die Backup-ZIP-Datei im selben Ordner, schließt den Kibana-Index und stellt den Index wieder her.

> **Information**: Das Skript *restore.cmd* löscht ALLE Dateien und Unterordner innerhalb des in *SNAPSHOT_REMOTE_LOCATION* angegeben Ordners! Andernfalls könnten zusätzliche Dateien die Wiederherstellung durcheinander bringen. Nutzen Sie für das Skript eigenständige Ordnerpfade in Elasticsearch.

2.2.2.4 Beispielskript ‚restore.cmd'

```
@ECHO OFF
REM This script is an example how to restore the ELO Analytics/kibana index
to a specific Analytics archive.
REM You may/should/must adapt it to your needs, environment and backup
strategy.
REM You need a copy of this file for another archive if you have multiple
archives.
REM
REM Be aware that this script is just plain simple and may not work
properly under certain instances!
REM
REM You may use, copy and change it according to your needs on your own
risk.
REM


REM Extract the Applications: src\curl.exe and 7zip.exe
REM Curl Download:
https://dl.uxnr.de/build/curl/curl_winssl_msys2_mingw64_stc/curl-
7.59.0/curl-7.59.0.zip
REM 7ZIP Download: http://www.7-zip.de/download.html

REM Define your server connection here
SET ES_USERNAME=admin
SET ES_PASSWORD=elo
SET PROTOCOL=http
SET ES_SERVER=eloanalytics.local
SET ES_PORT=9200

REM Name the ES snapshot configuration (any name)
SET ES_SNAPSHOT_CONFIG_NAME=analytics_backup
REM This snapshot (backup of kibana index) will be named :
SET ES_SNAPSHOT_NAME=automatic_snapshot_1
REM Set archive name here which defines the name of the kibana index to be
backed up
SET ELO_ARCHIVE_NAME=demo1

REM Where shall ES store its backup. This path must be allowed in
elasticsearch.yml
REM elasticsearch.yml
REM path.repo: ["C:\\BACKUP\\Analytics"]
REM
REM searchguard.enable_snapshot_restore_privilege: true
REM searchguard.check_snapshot_restore_write_privileges: false
SET ES_SNAPSHOT_LOCATION=C:\\BACKUP\\Analytics
REM The script can be run on another computer, so we need the location of
ES_SNAPSHOT_LOCATION on this computer if shared
REM WARNING: THE CONTENTS OF THIS FOLDER WILL BE ERASED!!
REM But if the script is run on the same computer just this (must be
absolute path):
```

```
REM SNAPSHOT_REMOTE_LOCATION=%ES_SNAPSHOT_LOCATION%
SET SNAPSHOT_REMOTE_LOCATION=\\eloanalytics.local\Analytics
REM ZIP Backup name and location
SET ZIP_LOCATION=backup.7z
REM 7ZIP command line arguments
SET ZIP_APP=7z x -y


REM ... HERE WE GO ...


SET FIRST_PART=curl
SET
ES_SERVER_PART=%PROTOCOL%://%ES_USERNAME%:%ES_PASSWORD%@%ES_SERVER%:%ES_POR
T%

SETLOCAL

echo.
echo Checking ES health
%FIRST_PART% -f %ES_SERVER_PART%%/_cat/health
IF ERRORLEVEL 1 (
  echo Error while calling:
  GOTO EXIT
)
ENDLOCAL

SETLOCAL

rd /s /q %SNAPSHOT_REMOTE_LOCATION%

echo. Unzip whole snapshot folder to file %ZIP_LOCATION%
%ZIP_APP% %ZIP_LOCATION% -o%SNAPSHOT_REMOTE_LOCATION%
IF ERRORLEVEL 1 (
  echo Error while calling:
  echo %ZIP_APP% %ZIP_LOCATION% -o%SNAPSHOT_REMOTE_LOCATION%
  GOTO EXIT
)

echo.
echo.
echo Closing Kibana Index

REM POST /.kibana_demo1/_close
%FIRST_PART% -XPOST %ES_SERVER_PART%/.kibana_%ELO_ARCHIVE_NAME%/_close -f
IF ERRORLEVEL 22 GOTO :NEXT_1
IF ERRORLEVEL 1 (
  echo Error while calling:
  echo %FIRST_PART% -POST
%ES_SERVER_PART%/.kibana_%ELO_ARCHIVE_NAME%/_close -f
```

```
  GOTO EXIT
)
:NEXT_1

echo.
echo Restoring Backup
REM POST /_snapshot/analytics_backup/snapshot_1/_restore
REM {
REM "indices": [".kibana_*"],
REM "ignore_unavailable": true,
REM "include_global_state": false,
REM "rename_pattern": "index_(.+)",
REM "rename_replacement": "restored_index_$1",
REM "include_aliases" : true,
REM "partial" : false
REM }
REM @ECHO ON
SET OPTIONS="-H \"Content-Type: application/json\" -d
\"{\"indices\":[\".kibana_%ELO_ARCHIVE_NAME%\"],\"ignore_unavailable\":fals
e,\"include_global_state\":false,\"rename_pattern\":\"index_(.+)\",\"rename
_replacement\":\"restored_index_$1\",\"include_aliases\":true,\"partial\":t
rue, \"include_aliases\":true}\""

%FIRST_PART% -XPOST
%ES_SERVER_PART%/_snapshot/%ES_SNAPSHOT_CONFIG_NAME%/%ES_SNAPSHOT_NAME%/_re
store?wait_for_completion=true %OPTIONS% -f
IF ERRORLEVEL 1 (
  echo Error while calling:
  echo %FIRST_PART% -XPOST
%ES_SERVER_PART%/_snapshot/%ES_SNAPSHOT_CONFIG_NAME%/%ES_SNAPSHOT_NAME%/_re
store?wait_for_completion=true %OPTIONS% -f

  %FIRST_PART% -XPOST %ES_SERVER_PART%/.kibana_%ELO_ARCHIVE_NAME%/_open -f
  GOTO EXIT
)

echo.
echo Waiting 10s for Completition
ping 127.0.0.1 -n 11 > nul

echo.
echo Show Kibanaindex Status
REM GET _cat/indices
%FIRST_PART% %ES_SERVER_PART%/_cat/indices -f | find /I
".kibana_%ELO_ARCHIVE_NAME%"
IF ERRORLEVEL 1 (
  echo Error while calling:
  echo %FIRST_PART% -XPOST
%ES_SERVER_PART%/.kibana_%ELO_ARCHIVE_NAME%/_open -f
  GOTO EXIT
)
```

```
ENDLOCAL

:EXIT
```

3 Weiterführende Literatur

- Snapshot-Dokumentation in der Elasticsearch
  https://www.elastic.co/guide/en/elasticsearch/reference/5.5/modules-snapshots.html
- Informationen zum globalen Status von Elasticsearch und dem Search-Guard-Index
  https://docs.search-guard.com/latest/snapshot-restore