ELSTER-*Transfer*

Handbuch Docker

ETR Version 26.01

Dokumentversion: 75

Veröffentlicht am 27.01.2026

Herausgeber: Bayerisches Landesamt für Steuern

Inhaltsverzeichnis

1 Systemvoraussetzungen

1.1 Betriebssystem

Für den Betrieb der ELSTER-Transfer-Anwendung als virtualisierter Container müssen folgende minimale Hardwareanforderungen gegeben sein:

- leistungsfähige 64-Bit-CPU mit Support für KVM-Virtualisierung
- leistungsfähige Netzwerkanbindung an das Internet *oder* das Behördennetzwerk NdB-VN (ehem. DOI-Netz)
- mind. 4 GB Arbeitsspeicher
- 20 GB freier Festplattenspeicher

Je nach Nutzungsweise, Mengengerüst und Aufbau der Containervirtualisierung sind Anpassungen (Erhöhungen) erforderlich

- bezüglich des benötigten Festplattenspeichers in Abhängigkeit der Menge an Dateien, die mit ELSTER-Transfer versendet und empfangen werden sollen
- bezüglich der Anzahl und Leistungsfähigkeit der Prozessoren sowie des verfügbaren Arbeitsspeichers beim gleichzeitigen Betrieb mehrerer Container

Die ELSTER-Transfer-Anwendung verwendet "Docker" als standardisiertes Container-Format. Die vorliegende Anleitung setzt daher eine vorhandene Docker-Installation voraus. Ausführliche Informationen zur Installation und zur Verwendung der "Docker"-Virtualisierungsplattform siehe offizielle Dokumentation von Docker.

Darüber hinaus können für den Betrieb der ELSTER-Transfer-Anwendung auch andere Containervirtualisierungsplattformen verwendet werden, die mit dem Docker-Container-Format kompatibel sind. Das Vorgehen zum Importieren, Konfigurieren, Starten und Stoppen des Containers gilt grundsätzlich analog, allerdings müssen die konkret auszuführenden Befehle an die jeweilige Containervirtualisierungsplattform angepasst werden.

1.2 Portfreischaltung

Neue Portfreischaltungen ab ETR 24.07
Ab ETR-Version 24.07 entfällt der Zugriff auf den bisherigen Server für die Datenabholung und wird ersetzt durch die Schnittstelle "ELSTER-Objektspeicher" unter den angegebenen URLs. Bitte prüfen Sie, ob die netzwerktechnische Konfiguration (Firewall-Freigaben etc.) angepasst werden muss, sodass der Zugriff auf die Infrastruktur des ELSTER-Objektspeichers fehlerfrei möglich ist.

Die ELSTER-Transfer-Anwendung benötigt Zugriff ins Internet auf Port **443**. Dabei werden folgende URLs aufgerufen:

- https://download.elster.de/...

Bei Datenabholung/-übermittlung via Internet:

- https://datenannahme1.elster.de/ERiC_IAS/ERiClet
- https://datenannahme1.elster.de/Elster2/serversnoop
- https://datenannahme3.elster.de/ERiC_IAS/ERiClet
- https://datenannahme3.elster.de/Elster2/serversnoop
- https://objektspeicher.elster.de/api/...

Bei Datenabholung/-übermittlung via Netze des Bundes (NdB-VN, ehem. DOI-Netz):

- https://datenannahme1.elster.doi-de.net/ERiC_IAS/ERiClet

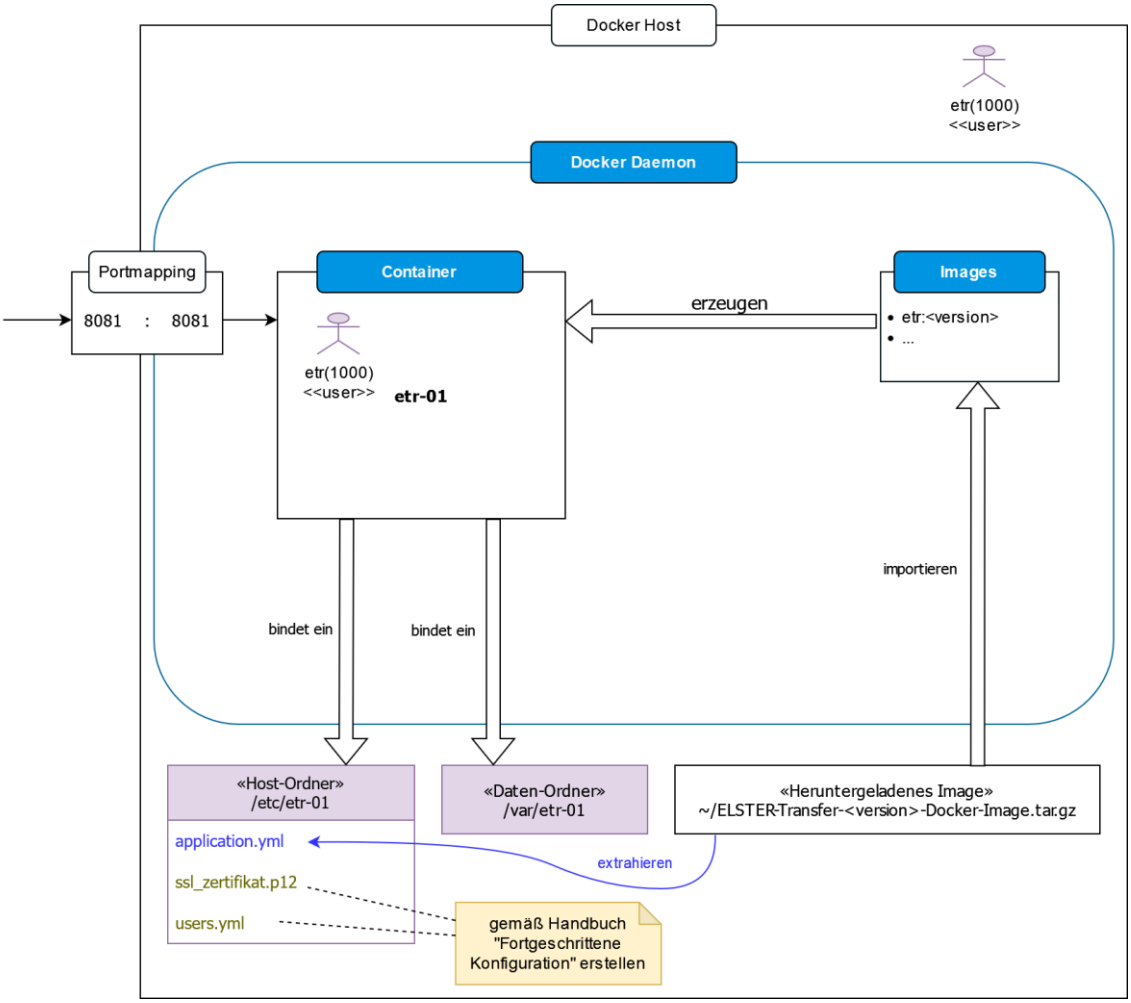
- https://datenannahme1.elster.doi-de.net/Elster2/serversnoop
- https://datenannahme2.elster.doi-de.net/ERiC_IAS/ERiClet
- https://datenannahme2.elster.doi-de.net/Elster2/serversnoop
- https://objektspeicher.elster.doi-de.net/api/...

In der "Elster4Konsens"-Testumgebung
(nur nach Aktivierung der Einstellung "etr.e4kAuswahl" in der Datei "application.yml" nutzbar und *nur für Testzwecke*!)

- https://datenannahme1-e4k.elster.de/ERiC_IAS/ERiClet
- https://datenannahme1-e4k.elster.de/Elster2/serversnoop
- https://datenannahme2-e4k.elster.de/ERiC_IAS/ERiClet
- https://datenannahme2-e4k.elster.de/Elster2/serversnoop
- https://datenannahme3-e4k.elster.de/ERiC_IAS/ERiClet
- https://datenannahme3-e4k.elster.de/Elster2/serversnoop
- https://datenannahme4-e4k.elster.de/ERiC_IAS/ERiClet
- https://datenannahme4-e4k.elster.de/Elster2/serversnoop
- https://objektspeicher-e4k.elster.de/api/...

2 Docker Image importieren und konfigurieren

Übersichtgrafik



2.1 Nutzer auf dem Host (und im Docker Container)

Auf dem Host wird ein dedizierter Nutzer `etr` erwartet. Dieser muss im Vorfeld angelegt werden. Das Container-Image ist so vorkonfiguriert, dass es mit der uid/gid `1000` gestartet wird. Daher bietet es sich an, den Nutzer **etr** auf dem Host mit der jeweiligen uid/gid anzulegen. Wird eine abweichende uid/gid für den Nutzer **etr** auf dem Host verwendet, muss Diese beim Start des Containers angegeben werden.

Über die gemeinsame uid/gid wird erreicht, dass die ETR-Anwendung, die im Docker Container läuft, im eingebundenen Host Verzeichnis mit konsistenten Lese-/Schreibberechtigungen arbeiten kann und auf dem Host sinnvolle Zugriffsberechtigungen gesetzt werden können.

2.2 Importieren des ELSTER-Transfer-Images

Laden Sie sich die komprimierte Imagedatei von der Downloadseite herunter und importieren Sie das Image in Ihren Docker-Daemon.

```
sudo docker load --input ELSTER-Transfer-${VERSION}-Docker-Image.tar.gz
```

2.3 Auspacken der Konfiguration auf dem Host

In der Imagedatei ist eine Konfigurationsdatei enthalten, die auf dem Host abgelegt wird, um bequem Anpassungen an ELSTER-Transfer vornehmen zu können:

```
# Konfigurationsverzeichnis anlegen / Nur bei Erstinstallation nötig
sudo mkdir /etc/etr-01

# Dateien auspacken
sudo tar -xvf ELSTER-Transfer-${VERSION}-Docker-Image.tar.gz \
  -C /etc/etr-01/ \
  application.yml secrets.env EXAMPLE-users.yml

# Zugriffsberechtigungen setzen (etr mit gid 1000)
sudo chown root:etr /etc/etr-01/application.yml \
  /etc/etr-01/EXAMPLE-users.yml
sudo chmod 640 /etc/etr-01/application.yml /etc/etr-01/EXAMPLE-users.yml
sudo chown root:root /etc/etr-01/secrets.env
sudo chmod 600 /etc/etr-01/secrets.env
```

2.4 Konfigurationsdateien auf dem Host anpassen

2.4.1 Remote-Zugriff

Um die größtmögliche Sicherheit über alle ETR-Installationen hinweg zu gewährleisten, ist standardmäßig nur der Zugriff vom lokalen System möglich. Daher müssen über einen regulären Ausdruck die IP-Adressen freigeschaltet werden, von denen auf ELSTER-Transfer zugegriffen werden soll. Dies geschieht in der extrahierten */etc/etr-01/application.yml.*

Die Angabe erfolgt in Form eines regulären Ausdrucks nach Java-Konvention.

```
etr:
  access:
    allowFrom: '<regex>'
```

Beispiele für den regulären Ausdruck (Platzhalter '<regex>' ersetzen):

- Zugriff nur über ein IPv4-Subnetz (10.1.0.0/16):
  '10\.1\.\d+\.\d+'
- Zugriff von 10.1.2.3 und allen IPv6-Adressen beginnend mit 2001:0db8:
  '10\.1\.2\.3|2001:0db8:.*'
- Alle Requests zulassen:
  '.*'

2.4.2 TLS-/SSL-Verschlüsselung und Client-Authentifizierung

Wir empfehlen dringend, die im → Handbuch „Fortgeschrittene Konfiguration“ beschriebenen Sicherheitsmaßnahmen für den "Server-Betrieb" bzw. "Remote-Zugriff" zu implementieren.

Dazu gehören

- die Konfiguration der TLS/SSL-Verschlüsselung mit einem entsprechenden Serverzertifikat
- die Konfiguration der Client-Authentifizierung, d.h. das Hinterlegen einer festen Liste mit zulässigen Benutzern in Form einer YAML-Datei, die sich gegenüber ETR per "Basic Auth"-Verfahren authentifizieren

- *je nach Anwendungsszenario:*
  - Einstellungen für den Betrieb mit einem Reverse Proxy
  - die Aktivierung des integrierten "Gruppen-Modus"

Damit die für diese Sicherheitsmaßnahmen zusätzlich erforderlichen Dateien (SSL-Zertifikat und YAML-Datei mit Benutzerinformationen) beim Start von ETR korrekt in den Docker-Container eingebunden werden, sollten sie ebenfalls im Konfigurationsordner liegen. Alternativ kann der Container um ein zusätzliches Verzeichnis- oder Volume-Mapping erweitert werden (Docker-Flags `–mount` oder `–v`).

Dateistruktur des Konfigurationsordners auf dem Hostsystem:

```
/etc/etr-01
 |- -rw-r----- root:etr  application.yml
 |- -rw------- root:root secrets.env
 |- -rw-r----- root:etr  ssl_zertifikat.p12
 |- -rw-r----- root:etr  users.yml
```

(Anmerkung: Das Serverzertifikat "ssl_zertifikat.p12" und YAML-Datei mit Benutzerinformationen zur Client-Authentifizierung "users.yml" können grundsätzlich frei / abweichend von diesem Beispiel benannt werden.)

In der "*application.yml*" werden die Pfade zu den jeweiligen Dateien abhängig von dem Pfad innerhalb des Containers angegeben.

```
server:
  ssl:
    key-store: /opt/ELSTER-Transfer/etr/config/mein_ssl_zertifikat.p12
```

Um den Zugriff von anderen Programmen auf die Dateien zu unterbinden, müssen die Berechtigungen entsprechend gesetzt werden.

```
sudo chown root:etr /etc/etr-01/mein_ssl_zertifikat.p12 /etc/etr-
01/users.yml
sudo chmod 640 /etc/etr-01/mein_ssl_zertifikat.p12 /etc/etr-01/users.yml
```

2.4.3 Nutzereinstellungen (ELSTER-Zertifikat) konfigurieren

Standardmäßig muss nach dem erstmaligen Start von ELSTER-Transfer das ELSTER-Zertifikatspasswort hinterlegt werden, bevor die Anwendung verwendet werden kann. Um bei einer Serverinstallation einen betriebsbereiten Container zur Verfügung stellen zu können, besteht ab ETR 3.1.0 die Möglichkeit, auch das ELSTER-Zertifikat bereits vor Start zu konfigurieren. Dafür muss das vorhandene ELSTER-Nutzerzertifikat im Konfigurationsordner */etc/etr-01* gespeichert werden. Allgemeine Informationen finden Sie hierzu im Handbuch „Fortgeschrittene Konfiguration“.

Beispielhafte Dateistruktur des Konfigurationsordners auf dem Hostsystem:

```
/etc/etr-01
 |- -rw-r----- root:etr  application.yml # Konfigurationsdatei
 |- -rw-r----- root:etr  elster-nutzerzertifikat.pfx # Beantragtes
Nutzerzertifikat für die Verwendung von ELSTER
 |- -rw------- root:root secrets.env # Notwendige Passwörter für die
Konfiguration
```

In der *application.yml* werden die Pfade zu den jeweiligen Dateien abhängig von dem Pfad innerhalb des Containers angegeben

```
etr:
  einstellungen:
    elsterZertifikat:
      dateiPfad: /opt/ELSTER-Transfer/etr/config/elster-
nutzerzertifikat.pfx
```

In der Datei /etc/etr-01/*secrets.env* wird das Passwort des Nutzerzertifikats hinterlegt:

```
ETR_EINSTELLUNGEN_ELSTERZERTIFIKAT_PASSWORT=<Zertifikatspasswort>
```

Hinweis zur Reihenfolge der Konfigurationsschritte
Ein neuer Docker-Container sollte immer erst nach dem Setzen aller Passwörter erstellt werden.
Sollte der Docker-Container bereits vorhanden sein (z.B. im Fall einer Update-Installation), **muss** bei Änderungen des Passworts der vorhandene Container entfernt und neu erstellt werden (wie unter "Erzeugen des Containers" beschrieben).

Um den Zugriff von anderen Programmen auf die Dateien zu unterbinden, müssen die Berechtigungen dementsprechend gesetzt werden:

```
sudo chown root:etr /etc/etr-01/elster-nutzerzertifikat.pfx
sudo chmod 640 /etc/etr-01/elster-nutzerzertifikat.pfx
```

2.4.4 Weitere Konfigurationen

Weiterführende Dokumentation zur *application.yml* finden Sie im Linux Handbuch sowie im Handbuch "Fortgeschrittene Konfiguration".

3 Erzeugen und Starten eines ETR-Containers

3.1 Erzeugen des Containers

Damit Dokumente und die interne Datenbank außerhalb des zustandslosen Containers gespeichert werden, muss das Datenverzeichnis auf dem Host erzeugt werden. Dies ist nur bei der Erstinstallation erforderlich. Bei einer Updateinstallation kann das bestehende Datenverzeichnis weiter verwendet werden. In dem Verzeichnis werden unter anderem die interne Datenbank, das ELSTER-Nutzer-Zertifikat und eingehende und ausgehende Dokumente gespeichert.

```
sudo mkdir /var/etr-01 # Datenverzeichnis erstellen
sudo chown etr:etr /var/etr-01  # Zugriffsrechte für das Datenverzeichnis
zuweisen.
```

Codeblock 1 Beispiel: Anlegen des ETR-Datenverzeichnisses

Zum Erzeugen und Starten des Containers muss das Datenverzeichnis nach `/home/etr/` und die Konfigurationsordner inklusive *application.yml* nach `/opt/ELSTER-Transfer/etr/config/` eingebunden werden.

Des Weiteren muss der Port 8081 des Containers mit der Option "*-p*" auch auf dem Host für den Zugriff von außerhalb freigeschaltet werden. Durch die Option '*restart unless-stopped*' wird der Container bei einem Server-Neustart automatisch mit dem Start des Docker-Daemons mitgestartet.
Wurde der Nutzer `etr` auf dem Host mit einer abweichenden uid/gid angelegt, so muss diese beim Erstellen des Containers angegeben werden.

```
sudo docker create -p 8081:8081 \
  --name etr-01 \
  --restart unless-stopped \
  -v /var/etr-01/:/home/etr/ \
  -v /etc/etr-01/:/opt/ELSTER-Transfer/etr/config/:ro \
  --env-file /etc/etr-01/secrets.env \
  etr:${VERSION}
```

Codeblock 2 Beispiel: Anlegen eines ETR-Containers in Docker mit Standardeinstellungen

Der Platzhalter "${VERSION}" muss dabei durch die konkrete ETR-Version ersetzt werden.

Bedeutung der Parameter:

- -p 8081:8081 – Der Port 8081 (Frei wählbarer Port) auf dem Docker-Host wird umgeleitet in den Container Port 8081 (Default-Port von ETR). Der Port auf dem Host ist der, unter dem ETR später aus dem Netzwerk erreichbar ist.
- --name – Frei wählbarer Name. Im Verlauf der Anleitung wird der Container mit 'etr-01' referenziert
- --restart – Container wird z. B. bei einem Neustart des Docker-Daemons automatisch mit neugestartet.
- -v /var/etr-01/:/home/etr - Einbinden des Datenverzeichnisses. Das angelegte Verzeichnis auf dem Host '/var/etr-01' wird innerhalb des Containers in den Pfad '/home/etr' eingebunden.
- -v /etc/etr-01/:/opt/ELSTER-Transfer/etr/config/:ro – Der Konfigurationsordner auf dem Host '/etc/etr-01' wird nur lesend innerhalb des Containers in den Pfad '/opt/ELSTER-Transfer/etr/config' eingebunden. Hier befinden sich alle notwendigen Dateien zur Konfiguration von ELSTER-Transfer.
- --env-file /etc/etr-01/secrets.env – Umgebungsvariablendatei, die u.a. zur sicheren Speicherung und Übergabe von Passwörtern in der Form "VARIABLE=WERT" verwendet

wird. Beim Erstellen des Containers liest der Docker-Daemon die Datei ein, um die definierten Umgebungsvariablen an ETR zu übergeben. Weitere Informationen zur Angabe der Passwörter finden Sie im Kapitel "Angabe von Passwörtern bei der Konfiguration" des Handbuchs "Fortgeschrittene Konfiguration".

- -u $(sudo -u etr id --user):$(sudo -u etr id --group) – Optional, falls die id/gid des Nutzers 'etr' auf dem Host-System von 1000:1000 abweicht.

Hinweis zur Reihenfolge der Konfigurationsschritte
Zum Zeitpunkt des Erstellen des Containers müssen die Passwörter bereits in der Datei *secrets.env* eingetragen sein. Ein neuer Docker-Container sollte daher erst nach Setzen aller Passwörter erstellt werden. Sollte der Docker-Container bereits vorhanden sein (z.B. im Fall einer Update-Installation), **muss** bei Änderungen eines Passworts der vorhandene Container entfernt und neu erstellt werden.

3.1.1 Konfiguration der Java-VM anpassen

Hinweis zur Verwendung von Nicht-Standard-Java-VM-Einstellungen
Anpassungen an der Konfiguration von ETR verwendete VM der Java Runtime (JRE) auf Nicht-Standardwerte wirken sich in der Regel stark auf das Performance-Verhalten der Anwendung aus und können daher die Funktion der Anwendung beeinträchtigen. Änderungen an dieser Stelle sollten nur im Ausnahmefall und nur mit sehr umfangreichen Erfahrungen im Betrieb von Java-basierten Anwendungen oder unter Anleitung des technischen Supports erfolgen.

Durch das Setzen der Umgebungsvariable "`JAVA_TOOL_OPTIONS`" können beispielsweise Einstellungen für das Speichermanagement verändert oder es können Optionen zur Fehleranalyse ergänzt werden.

Dazu wird beim Erzeugen des Docker-Containers die Umgebungsvariable "`JAVA_TOOL_OPTIONS`" entweder mit Hilfe des Parameters "`-e`" (bzw. "`--env`") direkt auf den angegebenen Wert gesetzt oder mit Hilfe des Parameters "`--env-file`" aus einer vorhandenen Umgebungsvariablendatei des Hosts eingelesen. Ohne Angabe eines entsprechenden Parameters gilt der hinterlegte Vorgabewert "`-Xms512m -Xmx1g`" (d.h. Größe des verwendeten "Heap"-Speichers initial 512 MiByte und maximal 1 GiByte).

3.2 Starten des Containers

Der Container kann wie folgt gestartet werden:

```
sudo docker start etr-01
```

3.3 Testen der Erreichbarkeit

Ob der Container erfolgreich hochgefahren wurde, kann durch Aufruf der URL http://localhost:8081/actuator/health/ping vom Hostsystem aus getestet werden. Erwartet wird das JSON-Objekt `{"status":"UP"}`.

```
curl http://localhost:8081/actuator/health/ping
```

Bevor Aufträge mit ELSTER-Transfer verarbeitet werden können, muss das ELSTER-Nutzer-Zertifikat über den Browser eingespielt werden, sofern dies nicht durch die Vorkonfiguration der Anwendung geschehen ist.

3.4 Container verwalten

3.4.1 Stoppen des Containers

Der Container kann wie folgt gestoppt werden:

```
sudo docker stop etr-01
```

3.4.2 Update des Containers

Um den ELSTER-Transfer-Container zu aktualisieren, muss zunächst der alte Container gestoppt und entfernt werden. Dabei bleiben der Daten-Ordner und die Konfigurationsdatei erhalten. Anschließend wird die neue Image-Version in Docker importiert.

```
sudo docker stop etr-01
sudo docker rm etr-01
sudo docker load --input ELSTER-Transfer-${VERSION}-Docker-Image.tar.gz
```

Des Weiteren müssen die folgenden Konfigurationsdateien aus dem bereitgestellten Image extrahiert werden (Details siehe Abschnitt „*Auspacken der Konfiguration auf dem Host*“), um sie jeweils manuell mit der vorhandenen Datei abzugleichen. Neu eingeführte Konfigurationsparameter aus der neuen ETR-Version müssen ggf. mit umgebungsspezifischen Werten ergänzt werden. Bestehende Konfigurationsparameter sollten anhand der erklärenden Kommentare zumindest geprüft und ggf. angepasst werden.

- `/etc/etr-01/application.yml`
- `/etc/etr-01/secrets.env`

Danach wird der Docker-Container mit der neuen ETR-Version wie im Abschnitt "Erzeugen des Containers" mittels `docker create` erzeugt (hierbei werden u.a. die Änderungen an der `secrets.env`-Datei ausgewertet). Der neu erstellte Container kann dann gestartet werden.

Abschließend kann das Image der vorherigen ETR-Version aus dem Docker-Daemon gelöscht werden:

```
sudo docker rmi etr:${OLD_VERSION} # Image der vorherigen Version
entfernen
```

Aktualisierung älterer Installationen bis einschließlich Version 3.4.0

Für ältere Versionen von ELSTER-Transfer bis einschließlich Version 3.4.0 ist die direkte Aktualisierung nicht möglich.

Stattdessen muss bei älteren Installationen zuerst eine Aktualisierung auf Version 24.07.1 durchgeführt werden. Diese Version sollte dann gestartet und auf korrekter Funktion geprüft werden. Wenn notwendig, erfolgt beim ersten Start automatisch eine Datenmigration (diese kann ggf. einige Minuten dauern).

Wenn die Version 24.07.1 erfolgreich installiert und fehlerfrei ausgeführt wird, kann anschließend durch das normale "Update des Containers" (siehe oben) eine weitere Aktualisierung auf die aktuelle ETR-Version erfolgen.

3.4.3 Entfernen des Containers

Das Entfernen ist erst möglich, nachdem der Container gestoppt wurde. Anschließend wird das importierte Image gelöscht. Der Daten-Ordner mit den Anwendungsdaten sowie die Konfigurationsdatei werden hierbei nicht gelöscht:

```
sudo docker stop etr-01 # Container stoppen
sudo docker rm etr-01 # Container entfernen
sudo docker rmi etr:${VERSION} # Image entfernen
```

Falls die Anwendungsdaten und die Konfigurationsdateien ebenfalls entfernt werden sollen, kann das Datenverzeichnis */var/etr-01/* und die der Konfigurationsordner */etc/etr-01/* gelöscht werden:

```
sudo rm -r /var/etr-01/ # Anwendungsdaten entfernen
sudo rm -r /etc/etr-01/ # Konfigurationsdateien entfernen
```

3.4.4 Containerstatus und Loginformationen

Eine Übersicht aller laufenden Container lässt sich durch `sudo docker ps` anzeigen.

Die Log-Ausgabe kann durch `sudo docker logs etr-01` ausgegeben werden, sofern zuvor die Ausgabe aktiviert worden ist.

Die Logdateien im Datenverzeichnis können vom Host aus eingesehen werden:

```
sudo cat /var/etr-01/elster-transfer/daten/log/elstertransfer.log
```

3.4.5 Einrichtung und Betrieb mehrerer ETR-Docker-Container

Mehrere Container mit unterschiedlichen ELSTER-Zertifikaten

Derzeit kann je ELSTER-Transfer-Instanz (Docker Container) nur 1 ELSTER-Zertifikat (ELSTER-Account) verwendet werden. Für verschiedene ELSTER-Transfer-Instanzen können verschiedene ELSTER-Zertifikate verwendet werden.

Wenn verschiedene ELSTER-Transfer-Instanzen auf einem Hostsystem ausgeführt werden sollen, wird je Instanz ein Konfigurationsordner und ein eigener Daten-Ordner benötigt. Zudem muss dem Container ein anderer Host-Port zugewiesen werden. Grundsätzlich sind die Namen des Containers und der Verzeichnisse frei wählbar. Entsprechend müssen die Aufrufe bei der Installation angepasst werden.

Ein Schema wäre, die Container zu nummerieren. Zum Beispiel: „*etr-01, etr-02, etr-03, ...“.* Im Folgenden muss „<index>“ durch eine Indexnummer ersetzt werden:

- Containername: etr-<index>
- Konfigurationsordner auf Host: /etc/etr-<index>
- Datenverzeichnis auf Host: /var/etr-<index>
- Host-Port beim Portmapping 808<index>:8081

Wird eine externe Datenbank (z.B. Postgres) verwendet, so muss jeder Container ein eigenes Datenbankschema erhalten.

Mehrere Container mit gleichem ELSTER-Zertifikat (Parallelbetrieb)

Die Verwendung mehrerer Container mit gleichem ELSTER-Zertifikat ist nur möglich, wenn bestimmte Annahmen erfüllt sind:

- Verwendung einer externen (Postgres-)Datenbank. Diese muss außerhalb ETR bereitgestellt und administriert werden. Alle Container teilen sich in diesem Fall die Datenbank / das gleiche Schema.
- Konfiguration der Einstellungen über die"application.yml"-Anwendungskonfigurationsdatei (Abschnitt "etr.einstellungen") und ggf. ergänzend Umgebungsvariablen, *nicht* über das "Einstellungen"-Formular der Weboberfläche (dieses dient dann nur zur Anzeige)
- Der Zugriff erfolgt primär über die REST-Schnittstelle (die Weboberfläche kann ggf. ergänzend zur Wartung und Debugging verwendet werden).
- Ein geteiltes Datenverzeichnis (z.B. ein gemapptes Volume oder Netzwerklaufwerk) ist vorhanden, auf das alle Instanzen zugreifen können.
- synchrone Systemuhrzeit (Abweichung deutlich kleiner als 1 Minute)
- *keine* "Sandbox"-Variante des Containers

Zwecks Debugging kann die Konfiguration des Anwendungslogfiles per "application.yml" (Schlüssel "logging.file.path") *instanz-spezifisch* auf unterschiedliche Ausgabeverzeichnisse angepasst werden, sodass jeder Knoten sein dediziertes Logfile schreibt. Die Angabe bezieht sich auf ein Verzeichnis innerhalb des jeweiligen Containers (kann ggf. per Verzeichnis- oder Volume-Mapping nach außen verfügbar gemacht werden).

3.4.6 Start des ETR-Browsers und Erstkonfiguration

Wenn der Container gestartet ist, können Sie die grafische Oberfläche von ETR auf Ihrem Webbrowser unter der URL http://localhost:8081/ aufrufen.

Weiterführende Dokumentation ist im Linux Handbuch zu finden.