

Toolbox for Research

Betrieb der Toolbox



mosaic-greifswald.de

Autoren: Martin Bialke, Arne Blumentritt, Ronny Schuldt

Version 1.3 vom 15.06.2017

mosaic@uni-greifswald.de

Inhalt

Allgemeines	2
Übersicht der mitgelieferten Skripte.....	3
Übersicht wichtiger Docker Kommandos.....	4
Integration eines eigenen Server-Zertifikats.....	4
Monitoren des Produktivsystems	4
Datenzugriff.....	5
Zugriff auf OpenClinica-Datensatzexport	6
Zugriff auf Export-Daten.....	6
Datensicherung und -wiederherstellung.....	6
Software-Updates.....	7
Formular-Updates	9

Allgemeines

Nach Installation der Toolbox muss die Einrichtung der Studie, der Standorte, des Erhebungsformulars und der Nutzer erfolgen. Lesen hierzu bitte das vorliegende Dokument *„Einrichtung der Toolbox“*.

Der Betrieb der Toolbox umfasst zahlreiche Aufgaben, wie die Datensicherung, das Verwalten der Passwörter u.v.m. Studienverantwortliche und technisches Personal lesen bitte dieses Dokument im Detail und ergänzend die *„Checkliste für Administratoren“*.

Rückfragen und Anmerkungen senden Sie bitte an mosaic@uni-greifswald.de.

Übersicht der mitgelieferten Skripte

Der Ordner toolbox_suppl enthält zahlreiche Skripte, die die Nutzung der Toolbox und die Steuerung der Container vereinfachen.

Skript	Zweck	Betroffene Komponenten der Toolbox
add_export_access_user.sh	Neuen Nutzer für Zugriff auf Web-Export hinzufügen	PORTAL
install.sh	Installation von Docker und der Toolbox for Research	Alle
install_oc.sh	Installation von OpenClinica (und Postgres)	OC, OC-DATA, OCDB
install_ocdi.sh	Installation und Aktualisierung von Data Dictionary, Forschungsdatenbank, der Integrations- und Exportprozesse	OCDI, MRDB, OCDI-DATA
install_portal.sh	Installation des Web-Portals	PORTAL, Zertifikate
install_ttp.sh	Installation des Dispatchers	TTP, TTPDB
set_passwords.sh	Setzen aller Passwörter	OC,TTP,OCDI
set_study_and_event.sh	Setzen der Default-Studie und der Default-Event-ID	TTPDB
start.sh	Starten der Toolbox	Alle
stop.sh	Stoppen der Toolbox	Alle
update_ocdi_burncase_dd.sh	Aktualisierung der BurnCase-Metadaten	OCDI, MRDB
uninstall.sh	WARNUNG VOLLSTÄNDIGE Deinstallation der Toolbox und LÖSCHUNG ALLER DATEN.	Alle

Tabelle 1 Übersicht mitgelieferter Skripte

Übersicht wichtiger Docker Kommandos

Zweck	Kommando
Laufende Container Anzeigen	<code>sudo docker ps</code>
Laufende Container Anzeigen inkl. DatenContainer	<code>sudo docker ps -a</code>
Container Logs anzeigen	<code>sudo docker logs <name></code>
Container Logs anzeigen, nur die aktuellsten 1000 Zeilen	<code>sudo docker logs --tail=1000 <name></code>
Container Konfiguration anzeigen	<code>sudo docker inspect <name></code>
Container starten	<code>sudo docker start <name></code>
Container stoppen	<code>sudo docker stop <name></code>
Container löschen	<code>sudo docker rm <name></code>
Container-Images anzeigen	<code>sudo docker images</code>
Container-Image löschen	<code>sudo docker rmi -f <image-id></code>
Alle Docker Images löschen (WARNUNG)	<code>sudo docker rmi -f \$(sudo docker images -q)</code>
Shell eines Containers aufrufen	<code>sudo docker exec -it <name> bash</code>
Datei aus Container herauskopieren	<code>sudo docker cp <containername>:<quellpfad im container> <zielpfad im hostsystem></code>
Stoppen aller System-Container	<code>sudo docker ps grep -v '\-data *\$' awk '{if(NR>1) print \$1}' xargs -r sudo docker stop</code>
Löschen aller gestoppten System-Container	<code>sudo docker ps -a -f status=exited grep -v '\-data *\$' awk '{if(NR>1) print \$1}' xargs -r docker rm</code>

Tabelle 2 Docker Kommando Übersicht

Integration eines eigenen Server-Zertifikats

- Per Putty den Portal-Container herunterfahren

```
sudo docker stop portal
```

- Per Winscp im Host-Verzeichnis `/opt/nginx/certs` das Zertifikat und den entsprechenden Schlüssel ablegen
- Per Putty im Host-Verzeichnis `/opt/nginx/conf` die Konfigurationsdatei entsprechend der Benennung der Zertifikat-Dateien anpassen
- Die Bezeichnung der Zertifikat-Dateien in der Konfigurationsdatei `/opt/nginx/conf/my_proxy.conf` aktualisieren und speichern
- Per Putty den Portal-Container neu starten

```
sudo docker start portal
```

Monitoren des Produktivsystems

- Logs eines Containers kontinuierlich anzeigen

```
sudo docker logs -f <containername>
```

- Nur die letzten 1000 Zeilen eines Container-Logs anzeigen

```
sudo docker logs -f --tail=1000 <containername>
```


Datenzugriff

- Anmeldedaten entnehmen Sie der Checkliste für Administratoren. Idealerweise sollten Sie diese Passwörter bereits per Skript geändert haben.

Data Dictionary und Forschungsdatenbank

- Per MySQL-Client über Port 3307 mit dem Toolbox-Server verbinden.
- Die Forschungsdaten werden in der Datenbank FODB gespeichert.
- Das Data Dictionary ist in der Datenbank G2D2 gespeichert.

TTP-Dispatcher

- Per MySQL-Client über Port 3306 mit dem Toolbox-Server verbinden.
- Die Dispatcher-Konfiguration ist in der Datenbank TTP_Dispatcher hinterlegt.
- Pseudonyme werden in der Datenbank GPAS verwaltet.
- Die Rollen- und Rechtevergabe zur Autorisierung der Depseudonymisierungsfunktionen wird in der Datenbank GRAS konfiguriert.

Zugriff auf OpenClinica-Datensatzexport

- Mit WinSCP via SFTP über Port 22 mit dem Toolbox-Server verbinden.
- In das Verzeichnis `/opt/extrains/original` navigieren.
- Nutzen Sie alternativ das Web-Frontend. Unter `<ihr server>/export/original` werden sämtliche OpenClinica-Export-Datensätze, die normalerweise über das OpenClinica-Frontend zu erreichen sind (HTML-Format, CSV, ...) zusätzlich abgelegt.
- **Achtung: Dies sind keine aggregierten Daten. Es sind keinerlei Gerätedaten, wie zum Beispiel BurnCase3D, enthalten.**

Zugriff auf Export-Daten

- Mit WinSCP via SFTP über Port 22 mit dem Toolbox-Server verbinden.
- In das Verzeichnis `/opt/extrains` navigieren.
- Nutzen Sie alternativ das Web-Frontend zu den Exporten. Der Zugriff ist per Basic Authentication abgesichert. Verwenden Sie das beigelegte Skript `add_export_access.sh` um neue Nutzer hinzuzufügen. Folgen Sie der Eingabeaufforderung. Der Portal-Container startet automatisch neu.

```
sudo ./add_export_access.sh
```

Datensicherung und -wiederherstellung

Die Datensicherung erfolgt auf Dateiebene. Die jeweiligen Docker-Container sollten für die Datensicherung gestoppt und nach Datensicherung wieder gestartet werden.

Sichern der OpenClinica-Produktivdaten

- Container für OpenClinica und OpenClinica-Datenbank stoppen und Rechte anpassen

```
sudo docker stop ocdb oc  
sudo chmod -R 777 /opt/openclinica/data/
```

- Den Ordner `/opt/openclinica/data` auf ein externes System sichern
- Rechte zurücksetzen und Container für OpenClinica-Datenbank, OpenClinica und Web-Portal starten

```
sudo chmod -R 700 /opt/openclinica/data/  
sudo docker start ocdb  
sudo docker start oc  
sudo docker restart portal
```

Sichern von MDR und Forschungsdatenbank

- Container für OCDI-Prozesse und entsprechende Datenbanken stoppen und Rechte anpassen

```
sudo docker stop ocdi  
sudo docker stop mrdp  
sudo chmod -R 777 /opt/mysql
```


- Den Ordner /opt/mysql/ocdi/data auf ein externes System sichern
- Container wieder starten

```
sudo docker start mrdp
sudo docker start ocdi
```

Sichern der TTP-Dispatcher Datenbanken

- Container für TTP-Dienst und TTP-Datenbanken stoppen und Rechte anpassen

```
sudo docker stop ttp
sudo docker stop ttpdb
sudo chmod -R 777 /opt/mysql
```

- Die Ordner /opt/mysql/thb/data auf ein externes System sichern
- Container neu starten

```
sudo docker start ttpdb
sudo docker start ttp
sudo docker restart portal
```

Zurückspielen der Produktivdaten

Verfahren Sie analog zur Datensicherung:

- Fahren Sie die entsprechenden Container herunter.
- Prüfen Sie ggf. die Schreib-/Leseberechtigungen der eingespielten Dateien.
- Spielen Sie die gewünschte Sicherung vom externen System in den angegebenen Ordner ein.
- Prüfen Sie ggf. die Schreib-/Leseberechtigungen der eingespielten Dateien.
- Starten Sie gestoppten Container in der oben genannten Reihenfolge erneut.
- Falls sich die Passwörter vom Zeitpunkt vor der Wiederherstellung und dem aktuellen Zeitpunkt unterscheiden, führen Sie bitte das Skript zum Ändern der Passwörter `set_passwords.sh` aus.
- Prüfen Sie, ob das System den Vorstellungen entsprechend funktioniert.

Software-Updates

Dispatcher-Modul

- Datensicherung durchführen
- Per WinScp verbinden mit Toolbox Server.
- Im Hostsystem in das Verzeichnis wechseln `opt/jboss-deploy`
- Die entsprechende EAR-Datei (Beispiel `nvr-dispatcher.ear`) löschen
- Orderansicht aktualisieren. Warte bis die Dateiendung der entsprechenden Datei vom Status „undeploying“ auf „undeployed“ wechselt.
- Entsprechende Datei mit Endung „undeployed“ abschließend ebenfalls löschen.
- Neue Version der EAR-Datei nach `opt/jboss-deploy` kopieren
- Deployment startet automatisch. Wird durch Wechsel der Datei-Endung auf „isdeploying“ und anschließend auf „deployed“ angezeigt.

OCDI-Modul

- Datensicherung durchführen
- Per WinScp verbinden mit Toolbox Server.
- Neuen OCDI-Ordner mit allen Bibliotheken und Skripten, sowie das aktuelle install_ocdi.sh Skript auf das Hostsystem (Home-Ordner) kopieren
- Benutzerpasswörter für **g2d2** und **fodb** in Skript **install_ocdi.sh** und Datei **/ocdi/routes.properties** anpassen
- Update Skript ausführbar machen und starten. Das Update dauert ca. 1 Minute. Dabei wird der OCDI-Container neu installiert und initialisiert. **Die Forschungsdatenbanken bleiben unverändert.**

```
sudo chmod 777 install_ocdi.sh
```

```
sudo ./install_ocdi.sh
```

- Nachdem der OCDI-Serverdienst sauber hochfahren konnte, einfach per STRG+C beenden und im Hintergrund laufen lassen

```
sudo start ocdi
```

Burncase-Metadaten-Beschreibung aktualisieren (im Falle neuer Variablen)

- Datensicherung durchführen
- Per WinScp verbinden mit Toolbox Server.
- Neues XLSX-Datei mit Burncase-Metadatenbeschreibung in OCDI-Unterordner kopieren (Skript update_ocdi_burncase_dd.sh und Metadaten-Datei sollten in gleichem Ordner liegen)
- Skript aufrufen und Metadatenbeschreibung für Burncase 2.6 aktualisieren

```
sudo chmod 777 update_ocdi_burncase_dd.sh
```

```
sudo ./update_ocdi_burncase_dd.sh DATEINAME.XLSX
```


Formular-Updates

- Neues Formular in OpenClinica einspielen
- Neues Event in OpenClinica für dieses Formular anlegen und neue Event_OID notieren
- Unique Protocol ID notieren (Tasks->Build Study->Create/View Study)
- Skript set_study_and_event.sh ausführen, das TTP-Modul startet automatisch neu
- Portal Modul neustarten

```
sudo docker restart portal
```

- Neues OpenClinica Export Dataset für das neue Formular erstellen (siehe Dokument „Einrichtung von OpenClinica“)
- Neuen Export-Job für das neue Dataset einrichten (siehe Dokument „Einrichtung von OpenClinica“)