

Installationsanleitung für Goobi.Production CE 1.10

Die folgende Anleitung beschreibt exemplarisch die Installation von Goobi.Production auf einem Debian-System mit lokaler Datenbank. Für andere Distributionen sind insbesondere die Pfadangaben entsprechend anzupassen. Soll ein separater Datenbankserver verwendet werden, ist dies beim Anlegen des Datenbanknutzers für Goobi bei der Rechtevergabe zu beachten (Schritt 3 der Grundinstallation).

Goobi.Production benötigt mindestens 1 leistungsstarke CPU und 2 GB RAM sowie ca. 10 GB Festplattenspeicher. Darin ist der Speicherplatzbedarf der Digitalisierungsdaten **nicht** enthalten!

Stand: Februar 2014. Kontakt und Rückfragen: Sebastian.Meyer@slub-dresden.de

Grundinstallation

1. Debian7 installieren

- OpenJDK7
- Tomcat7
- MySQL-Server 5.5

docker-dir/tomcat/Dockerfile
docker-dir/db/Dockerfile

Bei Goobi 1.x:

2. MySQL-Konfiguration für InnoDB anpassen (in /etc/mysql/my.cnf)

bind-address = 192.168.59.103 #
127.0.0.1
adr anpassen

- [mysqld]
- innodb_file_per_table
- MySQL-Dienst neustarten

docker-dir/db/my.cnf
- modified mysql config
- siehe [1]

3. MySQL-Datenbank und -Nutzer für Goobi anlegen

port = 3307

- mysql -uroot -p
- create database goobi;
- grant all privileges on goobi.* to goobi@localhost identified by 'goobi';
- flush privileges;
- exit;

gesetzt über:
docker-dir/docker-env.yml

4. Tomcat-Konfiguration für Speichermanagement anpassen (in /etc/default/tomcat7)

- JAVA_OPTS="-Djava.awt.headless=true -Xmx1920m -XX:MaxPermSize=128m -XX:+UseConcMarkSweepGC"
- Tomcat-Dienst neustarten

docker-dir/tomcat/catalina.sh

5. Goobi.Production CE-WAR-Datei von Github ins webapps-Verzeichnis des Tomcat kopieren

- Deployment läuft automatisch
- WEB-INF/classes/hibernate.cfg.xml anpassen
 - hibernate.connection.url
 - hibernate.connection.username
 - hibernate.connection.password
- Goobi initialisiert die Datenbank beim ersten Zugriff

Bei Goobi 1.x:

url setzen
statt localhost

docker-dir/tomcat/Dockerfile
docker-copmose.yml
- volumes

hibernate.cfg.xml ...
- in Goobi 2.0 nicht enthalten???

6. Administrator-Nutzer (admin/goobi) für Goobi anlegen

- Vorlage Goobi/setup/default.sql von Github herunterladen
- Dateikodierung beachten (muss UTF-8 sein)!
- mysql -ugoobi -Dgoobi --password=goobi < default.sql

docker-dir/db/default.sql
- auf mehrere Insert Skripte
aufgeteilt
docker-dir/db/Dockerfile
ADD Goobi_setup_schema.sql
ADD ...

Grundkonfiguration

- Verzeichnisse anlegen (Pfade nach Bedarf anpassen)

docker-dir/tomcat/Dockerfile
 RUN mkdir -p /usr/local/goobi/config/
 RUN chmod 777 /usr/local/goobi/config/
 ...

- /usr/local/goobi/config/ (Konfiguration)
- /usr/local/goobi/debug/ (Debug-Meldungen des OPAC-Beautifiers)
- /usr/local/goobi/import/
- /usr/local/goobi/logs/ (Log-Meldungen)
- /usr/local/goobi/messages/ (Lokalisierungsdateien)
- /usr/local/goobi/metadata/ (Vorgangsverzeichnisse)
- /usr/local/goobi/plugins/ (Funktionsmodule)
 - ./command/
 - ./import/
 - ./opac/
 - ./step/
 - ./validation/
- /usr/local/goobi/rulesets/ (Regelsätze)
- /usr/local/goobi/scripts/ (Shell-Skripte)
- /usr/local/goobi/swap/ (ausgelagerte Vorgänge)
- /usr/local/goobi/temp/ (Temporäre Dateien)
- /usr/local/goobi/users/ (Benutzer-Verzeichnisse)
- /usr/local/goobi/xslt/ (XSL-Stylesheets)

statt: /usr/local/goobi/
 dieses verwendet: /opt/digiverso/goobi/

- Zugriffsrechte für Tomcat-User (z.B. tomcat7) für die Verzeichnisse anpassen
 - Lese-/Schreibzugriff für grün markierte Verzeichnisse
 - Lesezugriff für alle anderen Verzeichnisse
- WEB-INF/classes/goobi_config.properties anpassen
 - Verzeichnisse konfigurieren

es darf jeder alles!
 chmod 777 ...

statt: /usr/local/goobi/
 dieses verwendet: /opt/digiverso/goobi/

- MetadatenVerzeichnis=/usr/local/goobi/metadata/
- RegelsaetzeVerzeichnis=/usr/local/goobi/rulesets/
- KonfigurationVerzeichnis=/usr/local/goobi/config/
- xsltFolder=/usr/local/goobi/xslt/
- dir_Users=/usr/local/goobi/users/
- debugFolder=/usr/local/goobi/debug/
- pluginFolder=/usr/local/goobi/plugins/
- swapPath=/usr/local/goobi/swap/
- tempfolder=/usr/local/goobi/temp/
- localMessages=/usr/local/goobi/messages/
- Verzeichnisse für Scans konfigurieren
 - DIRECTORY_SUFFIX=tif
 - DIRECTORY_PREFIX=orig
 - useOrigFolder=true (Orig-Verzeichnis wird als Standard verwendet)
 - createOrigFolderIfNotExists=true (Goobi legt Orig-Verzeichnis automatisch an)
- Visuelle Konfiguration
 - language.force-default=en
 - ApplicationHeaderTitle=Goobi
 - ApplicationTitle=Goobi.Production CE
 - ApplicationTitleStyle=font-size:17; font-family:verdana; color:white;
 - ApplicationLogo=goobi_meta_klein.jpg (in newpages/images/template/)
 - ApplicationHeaderBackground=goobi_meta_verlauf.jpg (in newpages/images/template/)
 - ApplicationWebsiteMsg=goobiWebseite (Key aus der messages-Datei)
 - ApplicationHomepageMsg=allgemeinesTextDemo (Key aus der messages-Datei)
 - ApplicationTechnicalBackgroundMsg=technischerHintergrundTextDemo (Key aus der messages-Datei)
 - ApplicationImpressumMsg=impressumTextDemo (Key aus der messages-Datei)
 - ApplicationIndividualHeader=Goobi Community

docker-dir/tomcat/Dockerfile
 ADD goobi_config.properties ...

steht so in der
 goobi_config.properties

steht so in der
 goobi_config.properties

steht so in der
 goobi_config.properties

wohin? newpages ???

docker-dir/tomcat/Dockerfile
COPY data/script_*.sh

steht so in der
goobi_config.properties

statt: /usr/local/goobi/scripts/
dieses: /opt/digiverso/goobi/scripts

number...=8

steht so in der
goobi_config.properties

leicht angepasst in der
goobi_config.properties

leicht angepasst in der
goobi_config.properties

steht so in der
goobi_config.properties

- ApplicationHomepageMsg=
- ApplicationVersionLogo= (URL der Logo-Datei)
- ApplicationWebsiteUrl= (Base-URL der Goobi-Installation)
- showStatisticsOnStartPage=true (Statistik auf Startseite anzeigen)
- Sicherheitseinstellungen
 - superadminpassword=098f6bcd4621d373cade4e832627b4f6 (MD5-Hash des Kennworts für Zugang zur Administrationsoberfläche)
 - anonymize=false (Benutzerinformationen in Statistik anonymisieren)
- Shell-Skripte konfigurieren (Vorlagen in Github unter **Goobi/scripts/**)
 - script_createDirUserHome=/usr/local/goobi/scripts/**script_createDirUserHome.sh** (Neues Benutzerverzeichnis anlegen)
 - script_createDirMeta=/usr/local/goobi/scripts/**script_createDirMeta.sh** (Neues Vorgangsverzeichnis anlegen)
 - script_createSymLink=/usr/local/goobi/scripts/**script_createSymLink.sh** (Vorgang im Home-Verzeichnis verlinken)
 - script_deleteSymLink=/usr/local/goobi/scripts/**script_deleteSymLink.sh** (Vorgang aus dem Home-Verzeichnis entfernen)
- Versionierung der Metadaten-Dateien:
 - formatOfMetaBackups=meta.*\\.xml.*+ **fehlt**
 - numberOfMetaBackups=8 (Anzahl der aufzuhebenden Versionen)
- Error-Handling
 - err_userHandling=true (Hinweistext zur Exception anzeigen)
 - err_linkToPage=../newpages/**statischTechnischerHintergrund.jsf** (Hinweislink bei Exception) **newpages ???**
 - err_emailEnabled=false (Ansprechpartner bei Exception anzeigen)
 - err_emailAddress1= (Adresse des ersten Ansprechpartners)
 - err_emailAddressX= (Adresse des X-ten Ansprechpartners)
- LDAP-Konfiguration
 - ldap_use=false
 - ldap_cert_root=/usr/local/goobi/config/**cacert.crt**
 - ldap_cert_pdc=/usr/local/goobi/config/**pdc.crt** **statt: /usr/local/goobi/config/
dieses: /opt/digiverso/goobi/config**
 - ldap_keystore_password=
 - ldap_keystore=/usr/local/goobi/config/**mykeystore.ke**
 - ldap_url=ldap://localhost:389/
 - ldap_nextFreeUnixId=cn=NextFreeUnixId,dc=goobi,dc=org **cn=NextFreeUnixId,dc=intranda,dc=com**
 - ldap_adminLogin=cn=Manager,dc=ldap,dc=goobi,dc=org **cn=Manager,dc=ldap,dc=intranda,dc=com**
 - ldap_adminPassword=
 - ldap_sslconnection=false
 - ldap_useTLS=false
 - ldap_readonly=true
 - ldap_encryption=SHA
 - ldap_attributeToTest=memberUid **fehlt**
 - ldap_valueOfAttribute=cn=goobi,cn=groups,dc=goobi,dc=org **fehlt**
- Module aktivieren
 - show_taskmanager=false (TaskManager in der Navigation einblenden)
 - show_modulmanager=false (ModulManager in der Navigation einblenden) **true**
 - goobiModuleServerPort=8000 (Port für die Modul-Kommunikation)
 - massImportAllowed=false (Massendaten-Import aktivieren; erfordert separates Modul) **true**
 - useWebApi=true (Schnittstelle zu „command“-Plugins aktivieren)
- Seitenbasierte OCR im Metadateneditor
 - showOcrButton=false (OCR im Editor aktivieren)
 - ocrUrl= (URL des OCR-Servers mit Parametern)
- Namenskonventionen
 - ImagePrefix=\\d{8} (Namenskonvention für Image-Dateien)

steht so in der
goobi_config.properties

steht so in der
goobi_config.properties

leicht angepasst in der
goobi_config.properties

steht so in der
goobi_config.properties

nicht gesetzt
(auskommentiert)

fehlen

wieder opt/diverso/goobi
ändern

docker-dir/tomcat/Dockerfile

COPY data/goobi_*.xml /opt/digiverso/goobi/config/
COPY data/laufzettel/*.xsl /opt/digiverso/goobi/xslt/

- ImageSorting=number (Numerische oder alphanumerische Sortierung)
- validateProzessTitelRegex=[\\w-]+ (Zulässige Zeichen für Vorgangstitel)
- Goobi ContentServer für PDF-Erzeugung
 - pdfAsDownload=true (PDF mit externem ContentServer erzeugen)
 - goobiContentServerUrl= (URL des externen Goobi-ContentServers)
 - goobiContentServerTimeout=30000 (Timeout für Aufrufe des Goobi-ContentServers)
- Metadateneditor konfigurieren
 - MetsEditorDisplayFileManipulation=false (Seiten in der Paginierungsansicht umsortieren) **fehlt**
 - MetsEditorDefaultSuffix=jpeg (Standardverzeichnis für Vorschaubilder)
 - MetsEditorDefaultPagination=uncounted (Standardpaginierungswert)
 - MetsEditorMaxTitleLength=0 (Titel gekürzt anzeigen)
 - MetsEditorEnableDefaultInitialisation=true (Standardwerte auch für Unterelemente setzen)
 - MetsEditorLockingTime=1800000 **fehlt**
 - useMetadatenvalidierung=true (Metadateneditor validiert gegen Regelsatz)
 - batchSize=500 (Maximale Anzahl angezeigter Vorgänge eines Batches)
- Konfiguration für DMS-Export
 - createSourceFolder=false
 - automaticExportWithImages=true (Image-Verzeichnisse exportieren)
 - automaticExportWithOcr=true (OCR-Verzeichnisse exportieren)
 - exportWithoutTimeLimit=true (Kein Timeout beim Export)
- ActiveMQ-Schnittstelle konfigurieren
 - activeMQ.hostURL=failover:(tcp://localhost:61616?closeAsync=false) (URL des ActiveMQ-Servers)
 - activeMQ.results.topic=GoobiProduction.ResultMessages.Topic
 - activeMQ.results.timeToLive=604800000 (entspricht 7 Tagen in Millisekunden)
 - activeMQ.createNewProcess.queue=GoobiProduction.CreateNewProcesses.Queue (Lese-Queue für Goobi.Production zum Anlegen eines Vorgangs)
 - activeMQ.finaliseStep.queue=GoobiProduction.FinaliseStep.Queue (Lese-Queue für Goobi.Production zum Beenden eines Arbeitsschritts)
- storageCalculationSchedule=-1 (automatische Ermittlung des Speicherbedarfs)
- useSwapping=false
- importUseOldConfiguration=false
- ContentServerUrl
- DatabaseAutomaticRefreshList
- DatabaseRefreshSessionWithoutUser
- DatabaseShareHibernateSessionWithUser
- doneDirectoryName
- ExportValidateImages
- MassImportUniqueTitle
- runHotfolder
- TiffHeaderArtists
- useLocalDirectory
- UserForImageReading
- useSimpleAuthentication
- WEB-INF/classes/log4j.properties anpassen:
 - log4j.appender.rolling.File=/usr/local/goobi/logs/goobi.log
 - log4j.logger.de.sub.goobi=ERROR, stdout, rolling
 - log4j.logger.org.goobi=ERROR, stdout, rolling
 - log4j.logger.ugh=ERROR, stdout, rolling
 - log4j.rootLogger=ERROR, stdout, rolling
- Alle goobi_*.xml und modules.xml aus WEB-INF/classes/ in das in goobi_config.properties angegebene Konfigurationsverzeichnis verschieben

Workflow-Problem:

- Tomcat startet nicht, mit bestehender DB
- ein restart ist nicht möglich
- Tabellen werden im nachhinein erstellt !!!
- daher per Insert über Datenbank

- `docket.xml` aus `WEB-INF/classes/` in das in `goobi_config.properties` angegebene `xmlFolder` verschieben
- Shell-Skripte unter `Goobi/scripts/*.sh` von Github an die in `goobi_config.properties` angegebenen Stellen (`script_*`) legen, `Ausführungsrechte` geben und ggf. anpassen `RUN chmod +x /opt/digiverso/goobi/scripts/*.sh`
- *Optional:* Verzeichnis `pages/imagesTemp/` in der WebApp als symbolischen Link auf ein temporäres Verzeichnis anlegen

nötig ???

- Damit Tomcat symbolischen Links folgt, muss auch dessen Konfiguration angepasst werden!
- Datei `goobi_projects.xml` im KonfigurationVerzeichnis anpassen
 - `<project name="default">` wird als Standard-Projekt verwendet, weitere Projekte können definiert werden
 - `<item>/<hide>` definiert das Mapping einzelner Metadatenfelder, wobei „hide“-Felder in der Oberfläche nicht angezeigt, aber dennoch prozessiert werden
 - `from="werk"` speichert Wert in Werkstückeigenschaften, `from="vorlage"` speichert Wert in Vorlageneigenschaften
 - `isdoctype/isnotdoctype` bestimmt, ob ein Feld für bestimmte Dokumenttypen angezeigt/nicht angezeigt werden soll; mehrere Dokumenttypen können durch „|“ getrennt werden
 - `ughbinding` bestimmt, ob der Wert in der `meta.xml` gespeichert wird
 - `metadata` bestimmt, in welches Metadatenfeld der Wert gespeichert werden soll
 - `required` definiert Pflichtfelder
 - `docstruct="topstruct"` speichert den Wert im obersten Strukturelement (z.B. mehrbändiges Werk), `docstruct="firstchild"` speichert den Wert im ersten Kind-Strukturelement (z.B. Band)
 - `<processtitle>` bestimmt die Bildungsvorschrift für Vorgangstitel
 - `<opac use="true">` definiert, ob ein Katalogimport möglich sein soll
 - `<catalogue>` gibt den Standard-Katalog aus `goobi_opac.xml` an
 - `<defaultdoctype>` gibt den voreingestellten Standard-Dokumenttyp aus `goobi_opac.xml` an
 - `<templates use="true"/>`
 - `<metadatageneration use="true"/>`
 - `<tifheader>` bestimmt die Bildungsvorschrift der TIF-Header
 - `<dmsImport/>`
 - `<validate>` bestimmt Validierungs- und Umformungsregeln für einzelne Metadatenfelder
 - `docstruct` (Regel trifft nur auf bestimmte Dokumenttypen zu)
 - `metadata` (Regel trifft nur auf bestimmtes Metadatenfeld zu)
 - `startswith` (Metadatenfeld beginnt mit dieser Zeichenfolge)
 - `endswith` (Metadatenfeld endet mit dieser Zeichenfolge)
 - `createelementfrom` (Namen der zusammenzuführenden Metadatenfelder)

aus Produktivsystem übernommen

???

- Datei `goobi_metadataDisplayRules.xml` im KonfigurationVerzeichnis anpassen
- Datei `goobi_processProperties.xml` im KonfigurationVerzeichnis anpassen
- Datei `goobi_digitalCollections.xml` im KonfigurationVerzeichnis anpassen

Plugins installieren

- OPAC-Import-Plugin
 - Datei `picaOpacImportClass.jar` aus Verzeichnis `Goobi/plugins/opac/` von Github herunterladen
 - Datei `picaOpacImportClass.jar` in das Unterverzeichnis `opac` des in `goobi_config.properties` angegebenen `pluginFolder` kopieren
 - `goobi_opac.xml` im KonfigurationVerzeichnis anpassen
 - Überprüfen, ob eigener Verbundkatalog in `<catalogue>` definiert ist
 - `<doctypes>` nach Bedarf um eigene Dokumenttypen aus PICA 0500/002@ ergänzen
 - `<beautify>` nach Bedarf um eigene Konditionen ergänzen

[1]

```
# my.cnf
[mysqld]
#
# * Basic Settings
#
innodb_file_per_table
!includedir /etc/mysql/conf.d/
```