

FU Berlin / MBPD
Installationsprotokoll Produktivsystem

Stand: 10.04.2014

Inhaltsverzeichnis

1	Installation.....	3
1.1	OS.....	3
1.2	Zusätzliche Packages.....	3
2	Konfiguration.....	4
2.1	sudoers einrichten (für das Deployment).....	4
2.2	Tomcat einrichten.....	4
2.2.1	Shell für User tomcat7 (für das Deployment).....	4
2.2.2	Spring-Applikations-Profil festlegen.....	4
2.3	PostgreSQL einrichten.....	4
2.3.1	Tablespace anlegen.....	4
2.3.2	Benutzer anlegen.....	4
2.3.3	Datenbank anlegen.....	5
2.4	Lucene vorbereiten.....	5
2.5	Blast und Schnittstelle installieren und konfigurieren.....	5
2.6	Konfigurationsdatei für Java-Anwendung anlegen.....	5
2.7	Backup.....	5

1 Installation

1.1 OS

Eine Standard-Installation Ubuntu 12.04 LTS wurde vorgefunden.

1.2 Zusätzliche Packages

```
apt-get install tomcat7 postgresql libpostgresql-jdbc-java apache2 \
    libapache2-mod-php5 openjdk-7-jre-headless
apt-get remove nano openjdk-6-*
```

2 Konfiguration

2.1 sudoers einrichten (für das Deployment)

Für das Deployment benötigt der Benutzer `tomcat7` die Möglichkeit, den Application-Server neu zu starten. Folgende Konfiguration erlaubt dies ohne zu weitgehende Rechte zu erteilen oder Passwörter verwalten zu müssen.

```
visudo -f /etc/sudoers.d/tomcat7
```

```
tomcat7    ALL = NOPASSWD: /etc/init.d/tomcat7
```

2.2 Tomcat einrichten

2.2.1 Shell für User `tomcat7` (für das Deployment)

Für das Deployment benötigt der Benutzer `tomcat7` eine Shell.

```
vim /etc/passwd
```

```
tomcat7:x:107:115::usr/share/tomcat7:/bin/bash
```

2.2.2 Spring-Applikations-Profil festlegen

```
vim /etc/default/tomcat7
```

Am Ende folgende Zeile anhängen:

```
JAVA_OPTS="$JAVA_OPTS -Dspring.profiles.active=live"
```

2.3 PostgreSQL einrichten

2.3.1 Tablespace anlegen

Daten sollen nicht im Standard-Pfad (`/var/lib/postgresql`), sondern unter `/data` gespeichert werden. Dafür muss ein eigener Tablespace erzeugt werden.

Als Benutzer `root`:

```
mkdir -p /data/mbpd/postgresql/mbpd_ts
chown -R postgres: /data/mbpd/postgresql
```

Als Benutzer `postgres` in der Datenbank-Shell (Programm `psql`):

```
create tablespace mbpd_ts location '/data/mbpd/postgresql/mbpd_ts';
```

2.3.2 Benutzer anlegen

Als user `postgres`:

```
createuser --no-createdb --no-inherit --login --no-creatorole --pwprompt \
--no-superuser mbpd
<2x passwd eingabe>
```

2.3.3 Datenbank anlegen

Als user postgres:

```
createdb --encoding=utf-8 --tablespace mbpd_ts --owner=mbpd mbpd \
"Mikrobiologische Primärdatenbank"
```

2.4 Lucene vorbereiten

Als user root:

```
mkdir /data/mbpd/lucene
chown tomcat7: /data/mbpd/lucene
```

2.5 Blast und Schnittstelle installieren und konfigurieren

Verzeichnisse anlegen:

```
mkdir /data/mbpd/blast
mkdir /data/mbpd/blast/db
chown -R www-data: /data/mbpd/blast
mkdir /data/mbpd/blast/bin
mkdir /data/mbpd/blast/transfer/
chown tomcat7: /data/mbpd/blast/transfer/
```

TODO Dateien kopieren, Pfade anpassen.

2.6 Konfigurationsdatei für Java-Anwendung anlegen

```
vim /var/lib/tomcat7/shared/classes/mbpd.properties
```

```
server.instance=F4D

labels.level1=Room
labels.level2=Box

database.user=mbpd
database.pass=<passwd aus 2.3.2 einsetzen>
database.url=jdbc:postgresql://127.0.0.1/mbpd

blast.url.search=http://127.0.0.1/search
blast.url.builddb=http://127.0.0.1/buildddb
blast.export=/tmp/blast.dat
blast.cron=0 0 1 1/1 * ? *

lucene.path=/data/mbpd/lucene
lucene.cron.cleanup=0 0/5 * 1/1 * ? *
```

2.7 Backup

```
vim /root/backup.sh
```

```
#!/bin/bash

#Variablen
error_log=/var/log/backup_error.log
postgresDump=/var/lib/postgresql/postgres_dump.sql

destination=/data/backup
destLucene=$destination/lucene/
destDb=$destination/db/postgres.sql
destBlast=$destination/blast/
destTomcatConfig=$destination/tomcat/config/
destTomcatWebapps=$destination/tomcat/webapps/
destApache=$destination/apache/

sourceLucene=/data/mbpd/lucene/*
sourceBlast=/data/mbpd/blast/*
sourceTomcatConfig=/var/lib/tomcat7/shared/classes/mbpd.properties
sourceTomcatWebapps=/var/lib/tomcat7/webapps/mbpd.war
sourceApache=/var/www/*

endFile=/data/backup/endfile
service=/usr/sbin/service
copy='/usr/bin/rsync -av --delete '
move='mv '

#Services stoppen
$service tomcat7 stop

#Dump Database
su - postgres -c "pg_dumpall > $postgresDump"

#Dateien kopieren
$copy $sourceLucene $destLucene 2>> $error_log
$copy $sourceBlast $destBlast 2>> $error_log
$copy $sourceTomcatConfig $destTomcatConfig 2>> $error_log
$copy $sourceTomcatWebapps $destTomcatWebapps 2>> $error_log
$copy $sourceApache $destApache 2>> $error_log
$move $postgresDump $destDb 2>> $error_log

#Create Endfile
touch $endFile

#Services starten
$service tomcat7 start
```

```
crontab -e
```

```
0 1 * * * sh /root/backup.sh
```