

Seafiler Server Migration von Ubuntu 14.04 LTS -> Debian 8 Jessie

Quelle:

1. https://manual.seafiler.com/maintain/backup_recovery.html
2. <https://unwahrscheinlichkeitsdrive.de/2015/02/07/seafiler-migration/>

Auf dem "alten" Server

```
<<Alle seafiler Dienste stoppen>>
```

```
mysqldump -h localhost -u root -p seafiler_ccnet-db > ccnet-db.sql  
mysqldump -h localhost -u root -p seafiler_seafiler-db > seafiler-db.sql  
mysqldump -h localhost -u root -p seafiler_seahub-db > seahub-db.sql
```

```
scp -r -P 22 /root/seafiler_db/ccnet-db.sql root@192.168.1.2:/root  
scp -r -P 22 /root/seafiler_db/seafiler-db.sql root@192.168.1.2:/root  
scp -r -P 22 /root/seafiler_db/seahub-db.sql root@192.168.1.2:/root
```

```
cd /home/seafiler/seafiler-server
```

```
tar -czvf - * | ssh -p 22 root@192.168.1.2 'tar -xhvf - -C /opt/seafiler/'
```

Auf dem "neuen" Server

```
apt install python2.7 libpython2.7 python-setuptools python-imaging \  
python-ldap python-mysqldb python-memcache python-urllib3 \  
memcached python-memcache sudo -y  
adduser --disabled-login --home /opt/seafiler --shell /bin/false --gecos  
"Seafiler" seafiler  
chown seafiler:root /opt/seafiler -R
```

```
mysql -u root -p
```

```
CREATE USER 'seafiler'@'localhost' IDENTIFIED BY 'PASSWORD1';
```

```
CREATE DATABASE `seafiler_ccnet-db`;
```

```
CREATE DATABASE `seafiler_seafiler-db`;
```

```
CREATE DATABASE `seafiler_seahub-db`;
```

```
GRANT ALL privileges ON `seafiler_ccnet-db`.* TO 'seafiler'@'localhost';
```

```
GRANT ALL privileges ON `seafiler_seafiler-db`.* TO 'seafiler'@'localhost';
```

```
GRANT ALL privileges ON `seafiler_seahub-db`.* TO 'seafiler'@'localhost';
```

```
flush privileges;
```

```
mysql -u root -p seafile_ccnet-db < /root/ccnet-db.sql
mysql -u root -p seafile_seafile-db < /root/seafile-db.sql
mysql -u root -p seafile_seahub-db < /root/seahub-db.sql

nano /opt/seafile/conf/ccnet.conf
nano /opt/seafile/conf/seafile.conf
nano /opt/seafile/conf/seahub_settings.py
nano /opt/seafile/ccnet/seafile.ini
su seafile
cd /opt/seafile/seafile-server-latest/
./seaf-fsck.sh
./seaf.sh start
./seahub.sh start-fastcgi
exit

mkdir /var/www/html/meinekleinefarm.net/seafile
nano /etc/apache2/sites-available/files.meinekleinefarm.net.conf
```

[files.meinekleinefarm.net.conf](#)

```
<VirtualHost *:80>
    ServerName files.meinekleinefarm.net
    Redirect permanent / https://files.meinekleinefarm.net/
</VirtualHost>

<IfModule mod_ssl.c>
<VirtualHost *:443>
    ServerAdmin webmaster@meinekleinefarm.net
    ServerName files.meinekleinefarm.net
    ServerAlias files.meinekleinefarm.net

    DocumentRoot /var/www/html/meinekleinefarm.net/seafile

    SSLEngine On
    SSLCertificateFile /opt/seafile/certificates/cacert.pem
    SSLCertificateKeyFile /opt/seafile/certificates/privkey.pem
    SSLCertificateChainFile
/opt/seafile/certificates/sub.class1.server.ca.pem

    Alias /media /opt/seafile/seafile-server-latest/seahub/media

<Location /media>
    ProxyPass !
    Require all granted
</Location>

    RewriteEngine On

    # seafile fileserver
    ProxyPass /seafhttp http://127.0.0.1:8082
```

```
ProxyPassReverse /seafhttp http://127.0.0.1:8082
RewriteRule ^/seafhttp - [QSA,L]

# seahub
SetEnvIf Request_URI . proxy-fcgi-pathinfo=unescape
SetEnvIf Authorization "(.*)" HTTP_AUTHORIZATION=$1
ProxyPass / fcgi://127.0.0.1:8000/
</VirtualHost>
</IfModule>
```

```
a2ensite files.meinekleinefarm.net.conf
service apache2 reload
```

Init Script

```
nano /etc/init.d/seafile-server
```

[seafile-server](#)

```
#!/bin/sh

### BEGIN INIT INFO
# Provides:          seafile-server
# Required-Start:    $local_fs $remote_fs $network mysql
# Required-Stop:     $local_fs
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Starts Seafile Server
# Description:       starts Seafile Server
### END INIT INFO

# Change the value of "user" to linux user name who runs seafile
user=seafile

# Change the value of "seafile_dir" to your path of seafile
# installation
# usually the home directory of $user
seafile_dir=/opt/seafile
script_path=${seafile_dir}/seafile-server-latest
seafile_init_log=${seafile_dir}/logs/seafile.init.log
seahub_init_log=${seafile_dir}/logs/seahub.init.log

# Change the value of fastcgi to true if fastcgi is to be used
fastcgi=true
# Set the port of fastcgi, default is 8000. Change it if you need
# different.
fastcgi_port=8000
```

```
#
# Write a polite log message with date and time
#
echo -e "\n\n About to perform $1 for seafiler at `date -Iseconds` \n "
>> ${seafiler_init_log}
echo -e "\n\n About to perform $1 for seahub at `date -Iseconds` \n "
>> ${seahub_init_log}

case "$1" in
    start)
        sudo -u ${user} ${script_path}/seafiler.sh ${1} >>
        ${seafiler_init_log}
        if [ $fastcgi = true ];
        then
            sudo -u ${user} ${script_path}/seahub.sh ${1}-
            fastcgi ${fastcgi_port} >> ${seahub_init_log}
        else
            sudo -u ${user} ${script_path}/seahub.sh ${1}
            >> ${seahub_init_log}
        fi
        ;;
    restart)
        sudo -u ${user} ${script_path}/seafiler.sh ${1} >>
        ${seafiler_init_log}
        if [ $fastcgi = true ];
        then
            sudo -u ${user} ${script_path}/seahub.sh ${1}-
            fastcgi ${fastcgi_port} >> ${seahub_init_log}
        else
            sudo -u ${user} ${script_path}/seahub.sh ${1}
            >> ${seahub_init_log}
        fi
        ;;
    stop)
        sudo -u ${user} ${script_path}/seahub.sh ${1} >>
        ${seahub_init_log}
        sudo -u ${user} ${script_path}/seafiler.sh ${1} >>
        ${seafiler_init_log}
        ;;
    *)
        echo "Usage: /etc/init.d/seafiler-server
        {start|stop|restart}"
        exit 1
        ;;
esac
```

```
mkdir -p /opt/seafiler/logs/
chmod +x /etc/init.d/seafiler-server
update-rc.d seafiler-server defaults
```

From:

<https://host.docker.internal:552/> - **Meine kleine Dokumentation**

Permanent link:

https://host.docker.internal:552/doku.php?id=seafire_server_migration&rev=1478128383

Last update: **2016/11/02 23:13**

