
Server Dokumentation Documentation

Release 1.0

Theodor van Nahl

Aug 17, 2018

Contents:

1	LDAP Server	1
1.1	Preface	2
1.2	Basis Setup	2
1.3	Administration	6
1.4	Replikations Setup	13
1.5	LDAP Client Anbindung	17
2	Indices and tables	21

- 389 Directory Server drop in replacement für OpenLDAP
- OSS Variante von Red Hat Directory Server
- Bei Fragen und Problemen kann die Red Hat Directory Server Dokumentation konsultiert werden

Die folgenden systemd-Units sind für dieses Kapitel relevant:

dirsrv.target Dieser target wird für alle `slapd`, also LDAP-Server Instancen, verwendet um sie zu kontrollieren.

dirsrv-admin.service Der Service dient der Administrator Konsole, bzw. der Serverkomponente dieser und ist nicht im `dirsrv.target` inbegriffen.

sssd.service Dieser Service läuft auf jedem Rechner der die Benutzer des LDAP auf dem lokalen System verfügbar macht. Der Dienst selbst ist ein Nachfolger von `nslcd` und vieler anderer Software die hier in ein System gebacken wurden.

Bei Problemen mit der Replikation sollte das [Kapitel 11.23ff](#) der Red Hat Directory Server 9 Dokumentation konsultiert werden.

Im folgenden noch einige Quellen die zur Erzeugung dieser Dokumentation konsultiert wurden:

- [389 Directory Server Dokumentation Übersicht](#)
- [Red Hat Directory Server 9 Admin Guide](#)
- [CentOS 5 Docs zu certutil](#)
- [certutil Zertifikatsimport von CA und Server-Zertifikat \(Linux Magazin\)](#)
- [389-Server SSL Aktivierung \(CentOS 5\)](#)
- [389ds Howto SSL](#)

1.1 Preface

Zunächst wurde auf den Servern **ishmael** und **shelob** die Datei */etc/hosts* angepasst, sodass die Server sich über das VPN Netz ansprechen.

1.2 Basis Setup

Der LDAP Server wird mit der folgenden Anleitung installiert:

```
root@shelob:~ # yum install 389-admin 389-ds-console
```

Anschließend wird ein Systemnutzer erzeugt unter dem der LDAP Server läuft:

```
root@shelob:~ # useradd -r -m -d /var/lib/ldap -s /sbin/nologin ldap
```

Nun kann das eigentliche Setup beginnen:

```
root@shelob:~ # setup-ds-admin.pl

=====
This program will set up the 389 Directory and Administration Servers.

It is recommended that you have "root" privilege to set up the software.
Tips for using this program:
  - Press "Enter" to choose the default and go to the next screen
  - Type "Control-B" then "Enter" to go back to the previous screen
  - Type "Control-C" to cancel the setup program

Would you like to continue with set up? [yes]:

=====
Your system has been scanned for potential problems, missing patches,
etc. The following output is a report of the items found that need to
be addressed before running this software in a production
environment.

389 Directory Server system tuning analysis version 14-JULY-2016.

NOTICE : System is x86_64-unknown-linux3.10.0-514.2.2.el7.x86_64 (2 processors).

NOTICE : The net.ipv4.tcp_keepalive_time is set to 7200000 milliseconds
(120 minutes). This may cause temporary server congestion from lost
client connections.

WARNING: There are only 1024 file descriptors (soft limit) available, which
limit the number of simultaneous connections.

WARNING : The warning messages above should be reviewed before proceeding.

Would you like to continue? [no]: yes

=====
Choose a setup type:

  1. Express
```

(continues on next page)

(continued from previous page)

Allows you to quickly set up the servers using the most common options and pre-defined defaults. Useful for quick evaluation of the products.

2. Typical

Allows you to specify common defaults and options.

3. Custom

Allows you to specify more advanced options. This is recommended for experienced server administrators only.

To accept the default shown in brackets, press the Enter key.

Choose a setup type [2]:

```
=====
Enter the fully qualified domain name of the computer
on which you're setting up server software. Using the form
<hostname>.<domainname>
Example: eros.example.com.
```

To accept the default shown in brackets, press the Enter key.

Warning: This step may take a few minutes if your DNS servers can not be reached or if DNS is not configured correctly. If you would rather not wait, hit Ctrl-C and run this program again with the following command line option to specify the hostname:

```
General.FullMachineName=your.hostname.domain.name
```

Computer name [shelob.lakrahn.de]:

```
=====
The servers must run as a specific user in a specific group.
It is strongly recommended that this user should have no privileges
on the computer (i.e. a non-root user). The setup procedure
will give this user/group some permissions in specific paths/files
to perform server-specific operations.
```

If you have not yet created a user and group for the servers, create this user and group using your native operating system utilities.

System User [dirsrv]:

System Group [dirsrv]:

```
=====
Server information is stored in the configuration directory server.
This information is used by the console and administration server to
configure and manage your servers. If you have already set up a
configuration directory server, you should register any servers you
set up or create with the configuration server. To do so, the
following information about the configuration server is required: the
fully qualified host name of the form
<hostname>.<domainname> (e.g. hostname.example.com), the port number
(default 389), the suffix, the DN and password of a user having
permission to write the configuration information, usually the
```

(continues on next page)

(continued from previous page)

configuration directory administrator, and if you are using security (TLS/SSL). If you are using TLS/SSL, specify the TLS/SSL (LDAPS) port number (default 636) instead of the regular LDAP port number, and provide the CA certificate (in PEM/ASCII format).

If you do not yet have a configuration directory server, enter 'No' to be prompted to set up one.

Do you want to register this software with an existing configuration directory server? [no]:

=====
Please enter the administrator ID for the configuration directory server. This is the ID typically used to log in to the console. You will also be prompted for the password.

Configuration directory server
administrator ID [admin]:
Password:
Password (confirm):

=====
The information stored in the configuration directory server can be separated into different Administration Domains. If you are managing multiple software releases at the same time, or managing information about multiple domains, you may use the Administration Domain to keep them separate.

If you are not using administrative domains, press Enter to select the default. Otherwise, enter some descriptive, unique name for the administration domain, such as the name of the organization responsible for managing the domain.

Administration Domain [lakrahn.de]:

=====
The standard directory server network port number is 389. However, if you are not logged as the superuser, or port 389 is in use, the default value will be a random unused port number greater than 1024. If you want to use port 389, make sure that you are logged in as the superuser, that port 389 is not in use.

Directory server network port [389]:

=====
Each instance of a directory server requires a unique identifier. This identifier is used to name the various instance specific files and directories in the file system, as well as for other uses as a server instance identifier.

Directory server identifier [shelob]:

=====
The suffix is the root of your directory tree. The suffix must be a valid DN. It is recommended that you use the dc=domaincomponent suffix convention. For example, if your domain is example.com, you should use dc=example,dc=com for your suffix.

(continues on next page)

(continued from previous page)

```

Setup will create this initial suffix for you,
but you may have more than one suffix.
Use the directory server utilities to create additional suffixes.

Suffix [dc=lakrahn, dc=de]:

=====
Certain directory server operations require an administrative user.
This user is referred to as the Directory Manager and typically has a
bind Distinguished Name (DN) of cn=Directory Manager.
You will also be prompted for the password for this user. The password must
be at least 8 characters long, and contain no spaces.
Press Control-B or type the word "back", then Enter to back up and start over.

Directory Manager DN [cn=Directory Manager]:
Password:
Password (confirm):

=====
The Administration Server is separate from any of your web or application
servers since it listens to a different port and access to it is
restricted.

Pick a port number between 1024 and 65535 to run your Administration
Server on. You should NOT use a port number which you plan to
run a web or application server on, rather, select a number which you
will remember and which will not be used for anything else.

Administration port [9830]:

=====
The interactive phase is complete. The script will now set up your
servers. Enter No or go Back if you want to change something.

Are you ready to set up your servers? [yes]:
Creating directory server . . .
Your new DS instance 'shelob' was successfully created.
Creating the configuration directory server . . .
Beginning Admin Server creation . . .
Creating Admin Server files and directories . . .
Updating adm.conf . . .
Updating admpw . . .
Registering admin server with the configuration directory server . . .
Updating adm.conf with information from configuration directory server . . .
Updating the configuration for the httpd engine . . .
Starting admin server . . .
The admin server was successfully started.
Admin server was successfully created, configured, and started.
Exiting . . .
Log file is '/tmp/setupeWqRaf.log'

```

Obwohl der Service nach dem Setup bereits läuft muss er dennoch für zukünftige Starts aktiviert werden und auch die Firewall muss geöffnet werden. Dabei soll der unverschlüsselte Ldap-Zugriff nur während der Einrichtung benutzbar sein:

```
root@shelob:~ # systemctl enable dirsrv.target
```

(continues on next page)

(continued from previous page)

```

root@shelob:~ # systemctl start dirsrv-admin.service
root@shelob:~ # firewall-cmd --add-port=9830/tcp --zone=internal
success
root@shelob:~ # firewall-cmd --add-service=ldap --zone=internal
success
root@shelob:~ # firewall-cmd --add-service=ldaps --zone=internal
success
root@shelob:~ # firewall-cmd --add-port=9830/tcp --zone=internal \
--permanent
success
root@shelob:~ # firewall-cmd --add-service=ldap --zone=internal \
--permanent
success
root@shelob:~ # firewall-cmd --add-service=ldaps --zone=internal \
--permanent
success

```

1.3 Administration

Für die Administration ist die 389-console notwendig die auf den Rechner des Administrators installiert werden muss. Unter CentOS ist dazu das Paket 389-console notwendig. Das Programm wird wie folgt installiert:

```
$ sudo yum install 389-console
```

Und wie folgt gestartet:

```
$ 389-console
```

Einen zugehörigen Menüeintrag habe ich nicht finden können.

1.3.1 LDAPS (und HTTPS für console)

Um Zertifikate auf dem LDAP Server zu managen wurde auf ishmael eine neue easy-rsa angelegt. Unter dieser CA müssen alle Server wie Clients ihre persönlichen Zertifikate erhalten. Die folgenden Schritte wurden daher auf Ishmael ausgeführt:

```

root@ishmael:~ # git clone https://github.com/OpenVPN/easy-rsa.git ldap-easy-rsa
root@ishmael:~ # cd ldap-easy-rsa/easyrsa3
root@ishmael:... # cp vars.example vars
root@ishmael:... # vim vars
84,88c84,88
< #set_var EASYRSA_REQ_COUNTRY      "US"
< #set_var EASYRSA_REQ_PROVINCE     "California"
< #set_var EASYRSA_REQ_CITY         "San Francisco"
< #set_var EASYRSA_REQ_ORG          "Copyleft Certificate Co"
< #set_var EASYRSA_REQ_EMAIL        "me@example.net"
---
> set_var EASYRSA_REQ_COUNTRY       "DE"
> set_var EASYRSA_REQ_PROVINCE      "Niedersachsen"
> set_var EASYRSA_REQ_CITY          "Salzgitter"
> set_var EASYRSA_REQ_ORG           "Lakrahn Group"
> set_var EASYRSA_REQ_EMAIL         "admin@lakrahn.de"

```

(continues on next page)

(continued from previous page)

```

root@ishmael:... # ./easymrsa init-pki
root@ishmael:... # ./easymrsa build-ca

Note: using Easy-RSA configuration from: ./vars
Generating a 2048 bit RSA private key
.....
<..+++
.....+++
writing new private key to '/root/ldap-easy-rsa/easymrsa3/pki/private/ca.key.hdt8N9TfVS
<'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Common Name (eg: your user, host, or server name) [Easy-RSA CA]: LDAP Lakrahn CA

CA creation complete and you may now import and sign cert requests.
Your new CA certificate file for publishing is at:
/root/ldap-easy-rsa/easymrsa3/pki/ca.crt

```

Damit die CA von dem Rechner auch akzeptiert wird ist es notwendig sie wie folgt zu installieren (geschrieben wurde die *pki/ca.crt* von ishmael):

```

root@shelob:~/ # vim /etc/pki/ca-trust/source/anchors/ldap-ca.crt
root@shelob:~/ # update-ca-trust extract

```

Der Management-Server benötigt einen Zertifikatsspeicher der über das Terminal angelegt werden muss. In diesem werden die Zertifikate für den HTTPS Zugang verwaltet. Da ich mit der Grafischen Oberfläche Probleme hatte (Passwort funktioniert entweder in der GUI xor CLI). Zunächst muss also der Passwortspeicher erzeugt werden:

```

root@shelob:~ # vim pwdfilename.txt
my_pwgen_password
root@shelob:~ # certutil -N -d /etc/dirsrv/slapd-shelob -f pwdfilename.txt

```

Woraufhin ein Zertifikatsrequest für den Server angelegt werden kann.

```

root@shelob:~ # certutil -R -k rsa -g 2048 \
-s "CN=shelob.lakrahn.de,L=Niedersachsen,ST=DE" \
-d /etc/dirsrv/slapd-shelob/ -a -o shelob.req
Enter Password or Pin for "NSS Certificate DB":

A random seed must be generated that will be used in the
creation of your key. One of the easiest ways to create a
random seed is to use the timing of keystrokes on a keyboard.

To begin, type keys on the keyboard until this progress meter
is full. DO NOT USE THE AUTOREPEAT FUNCTION ON YOUR KEYBOARD!

Continue typing until the progress meter is full:

```

(continues on next page)

(continued from previous page)

```
| ***** |
Finished.  Press enter to continue:

Generating key.  This may take a few moments...
```

Dieser Request Text muss anschließend auf dem Server mit der CA (in diesem Falle ishmael) unter z.B. `pki/reqs/shelob.req` abgelegt werden. Dieser Request kann dann vom Server mit dem folgenden Befehl signiert werden:

```
root@shelob:~/ldap-easy-rsa/easyrsa3 # ./easyrsa sign-req server shelob
```

Das Zertifikat daraus entstandene Zertifikat kann anschließend auf dem LDAP Server wie folgt importiert werden wie auch die CA:

```
root@shelob:~ # certutil -A -n "shelob-server-cert" -t "u,u,u" \
-i shelob.crt \
-d /etc/dirsrv/slapd-shelob/
Notice: Trust flag u is set automatically if the private key is present.
root@shelob:~ # certutil -A -n "LDAP CA" -t "CT,," -i ca.crt \
-d /etc/dirsrv/slapd-shelob/
```

Um letztendlich TLS und HTTPS zu aktivieren geht man in der 389-Console des LDAP-Servers über den *Configuration* Tab auf das Wurzelement und wählt dort den Untertab *Encryption*, wählt dort das Zertifikat aus und stellt alle Cipher bis auf **TLS/AES 256** ab (siehe *TLS Konfiguration in der 389-Console*).

Damit der Server anschließend den Passwort-Speicher öffnen kann ist es noch notwendig eine Datei mit der PIN anzulegen:

```
root@shelob:~ # cd /etc/dirsrv/slapd-shelob/
root@shelob:~/slapd-shelob # touch pin.txt
root@shelob:~/slapd-shelob # chmod 600 pin.txt
root@shelob:~/slapd-shelob # chown dirsrv:dirsrv pin.txt
root@shelob:~/slapd-shelob # vim pin.txt
Internal (Software) Token:my_pwgen_password
root@shelob:~/slapd-shelob # restorecon -v /etc/dirsrv/slapd-shelob/pin.txt
```

Um auch in der `dirsrv-console` HTTPS zu aktivieren muss zunächst in der Datei `/etc/dirsrv/admin-serv/nss.conf` die folgende Änderung vorgenommen werden.:

```
48c48
< NSSPassPhraseDialog builtin
---
> NSSPassPhraseDialog file:///etc/dirsrv/admin-serv/password.conf
```

Und die Datei `/etc/dirsrv/admin-serv/password.conf` mit dem im folgenden beschriebenen Inhalt (Groß- bzw. Kleinschreibung beachten!):

```
root@shelob:~ # cd /etc/dirsrv/admin-serv
root@shelob:/etc/dirsrv/admin-serv/ # touch password.conf
root@shelob:/etc/dirsrv/admin-serv/ # chmod 600 password.conf
root@shelob:/etc/dirsrv/admin-serv/ # chown dirsrv:dirsrv password.conf
root@shelob:/etc/dirsrv/admin-serv/ # vi password.conf
internal (software) token:mypassword
```

(continues on next page)

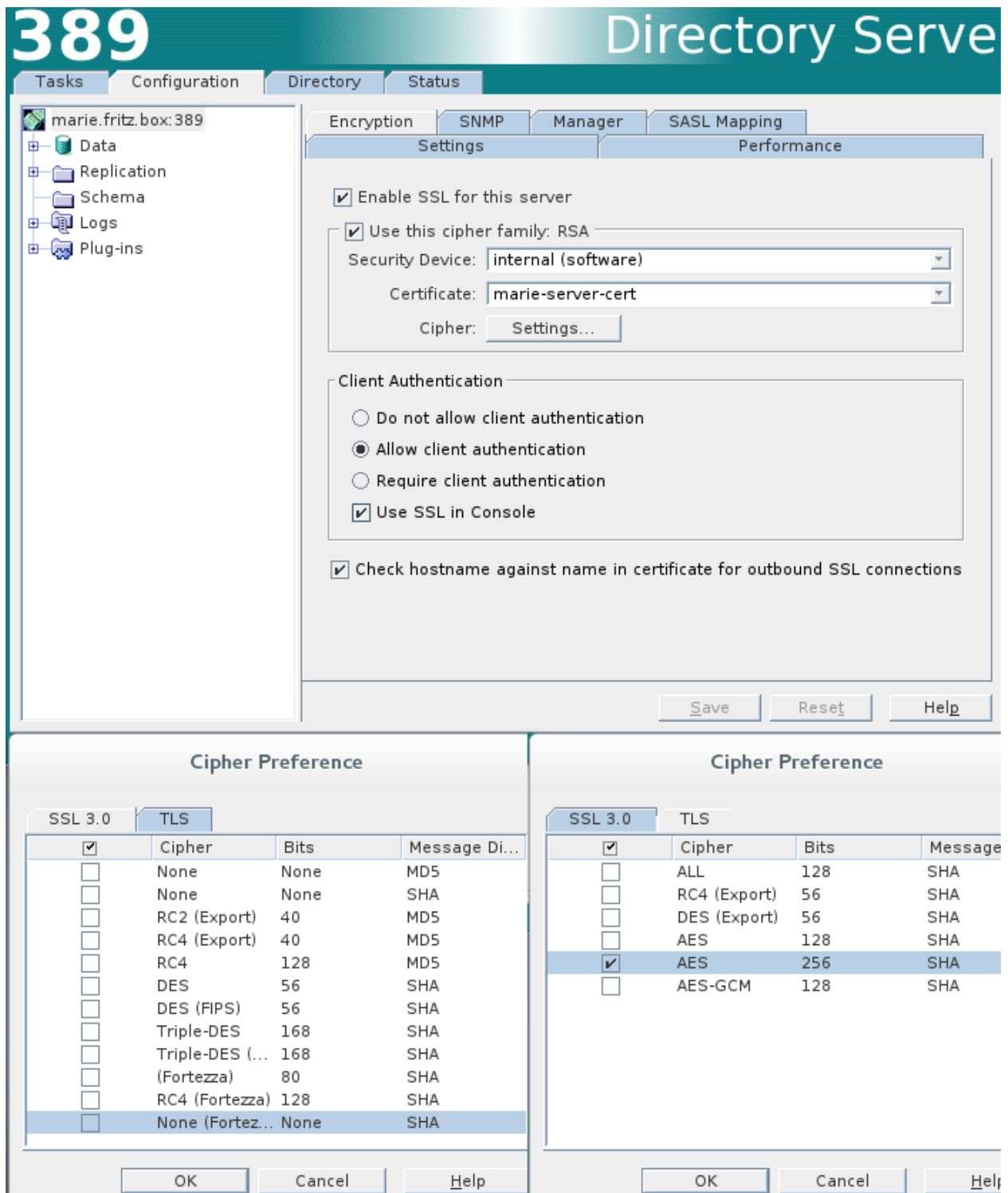


Fig. 1: TLS Konfiguration in der 389-Console

(continued from previous page)

```
root@shelob:/etc/dirsrv/admin-serv/ # restorecon -v password.conf
root@shelob:/etc/dirsrv/admin-serv/ # vi adm.conf
8c8
< ldapurl: ldap://shelob.lakrahn.de:389/o=NetscapeRoot
---
> ldapurl: ldaps://shelob.lakrahn.de:636/o=NetscapeRoot
```

Und an der Datei */etc/dirsrv/admin-serv/console.conf* folgende Anpassungen:

```
92c92
< NSSEngine off
---
> NSSEngine on
96c96
< NSSNickname server-cert
---
> NSSNickname shelob-server-cert
102c102
< NSSCertificateDatabase /etc/dirsrv/admin-serv
---
> NSSCertificateDatabase /etc/dirsrv/slapd-shelob/
```

Außerdem muss der Server den verbindenden Admin-Consolen mitteilen, dass jede Kommunikation verschlüsselt stattfinden soll. Dazu legen wir die Datei */tmp/serverSecurity.ldif* mit folgendem Inhalt an.

```
dn: cn=configuration,cn=admin-serv-shelob,cn=389 Administration Server,
    cn=Server Group,cn=shelob.lakrahn.de,ou=lakrahn.de,o=NetscapeRoot
changetype: modify
add: nsServerSecurity
nsServerSecurity: on
```

Diese Datei übernehmen wir mit folgendem Befehl in die Datenbank:

```
root@shelob:~ # ldapmodify -x -D "cn=Directory Manager" -W -f /tmp/serverSecurity.ldif
Enter LDAP Password:
modifying entry "cn=configuration,cn=admin-serv-shelob,cn=389 Administration Server,
↪ cn=Server Group,cn=shelob.lakrahn.de,ou=lakrahn.de,o=NetscapeRoot"
```

Zum Abschluss sollte der LDAP-Server wie auch der Admin-Server einmal neu gestartet werden um die Änderungen wirksam zu machen:

```
root@shelob:~ # systemctl restart dirsrv.target dirsrv-admin.service
```

Jetzt sollte sowohl der LDAP-Server via LDAPS und der Admin-Server via HTTPS erreichbar sein. Das kann schnell mithilfe von `openssl` getestet werden:

```
root@shelob:~ # echo QUIT | openssl s_client -connect localhost:ldaps && echo LDAPS_
↪Erfolgreich
...
LDAPS Erfolgreich
root@shelob:~ # echo QUIT | openssl s_client -connect localhost:9830 && echo HTTPS_
↪Erfolgreich
...
HTTPS Erfolgreich
```

1.3.2 Anonymous Binds

Im folgenden soll ein Element des LDAP-Servers vorgestellt werden, dass durch ownCloud nicht geändert werden kann. Es ist möglich sich auf den LDAP-Server zu verbinden und `dc=lakrahn,dc=de` abzufragen **ohne** sich in irgendeiner Weise zu authentifizieren. Das beinhaltet nicht die Passwort-Felder aber fast alle anderen Informationen der Nutzer wie Name, Uid, Gruppen, EMail und so weiter. Ausprobiert werden kann man das mit dem folgenden Befehl:

```
# ldapsearch -x -b "dc=lakrahn,dc=de"
```

Aus diesem Grund wird auch derzeit `sssd` mittels Anonymous-Login an die LDAP-Server angebunden. Sollte sich der Zustand bei ownCloud/NextCloud irgendwann ändern kann man das Verhalten wie folgt deaktivieren:

```
# ldapmodify -x -D "cn=Directory Manager" -w secret -h server.example.com -p 389
```

```
Enter LDAP Password:
dn: cn=config
changetype: modify
replace: nsslapd-allow-anonymous-access
nsslapd-allow-anonymous-access: rootdse
```

Zusätzlich müsste dann noch ein Authentifizierungsmechanismus via Zertifikate für die Clients konfiguriert werden damit die zulässigen Clients weitere Informationen abfragen können. Solange der aktuelle Stand jedoch besteht sollte darauf geachtet werden, dass das LDAP nur innerhalb des VPN-Netzes verfügbar ist und keine Queries über das Internet erlaubt werden.

1.3.3 Passwort Hashes

Als Default Hashing Algorithmus kommt SSHA (also SHA1 mit Salt) zum Einsatz. Da das nicht mehr Zeitgemäß ist muss auf dem Directory Server die Passwort Policy angepasst werden. Dazu öffnet man die 389 Konsole mit dem Nutzer `cn=Directory Manager` und wählt bei der Verwaltung den Directory Server (`shelob`) aus.

Auf `Shelob` bekommt man von der Console mehrere Tasks angeboten, die gerade nicht interessant sind. Unter dem *Configuration* Tab kann man aber direkt in der *Data* Sektion eine Passwort Policy festlegen. Dort wählt man als Hashing Algorithmus dann **SSHA512** (Salted SHA512; siehe [LDAP Passwort Policy Einstellungen in 389-Console](#).) und weiter wurde der Account Lockout eingerichtet, der bei 4 fehlerhaften Loginversuchen innerhalb von 10 Minuten den Account für 60 Minuten sperrt (siehe [LDAP Lockout Policy Einstellungen in der 389-Console](#).).

1.3.4 LDAP Backups

Es gibt mehrere Möglichkeiten von den Daten der LDAP-Datenbank backups anzulegen. Die einfachste Möglichkeit ist die 389-console zu verwenden und innerhalb der GUI des LDAP-Servers die Option *Back Up Directory Server* auszuwählen. Eine Wiederherstellung dieses Backups geht genauso einfach. Dabei werden die Backups unter `/var/lib/slaped-shelob/bak/` abgelegt.

Da die GUI keine Automatisierten Backups ermöglicht sollen hier einmal die drei Möglichkeiten via CLI besprochen werden und eine der Varianten im Detail vorgestellt werden. Die folgenden Informationen entstammen dem Red Hat Directory Server Administration Guide [Kapitel 4.3](#):

db2bak Diese Möglichkeit hat den Nachteil, dass sie nur bei ausgeschaltetem LDAP Server verwendet werden kann. Das Skript sollte nicht bei einem laufenden LDAP Server eingesetzt werden. Dafür bieten die hiermit angelegten Backups den Vorteil, dass sie bei einem ausgeschalteten LDAP-Server wieder eingespielt werden müssen. Sollte im Netzwerk die LDAP Replikation verwendet werden vereinfacht diese Variante das Einspielen von alten Backups. Sollte jedoch eine Single-Master Replikation eingesetzt werden wird von diesem Verfahren auf dem Master-Knoten abgeraten.

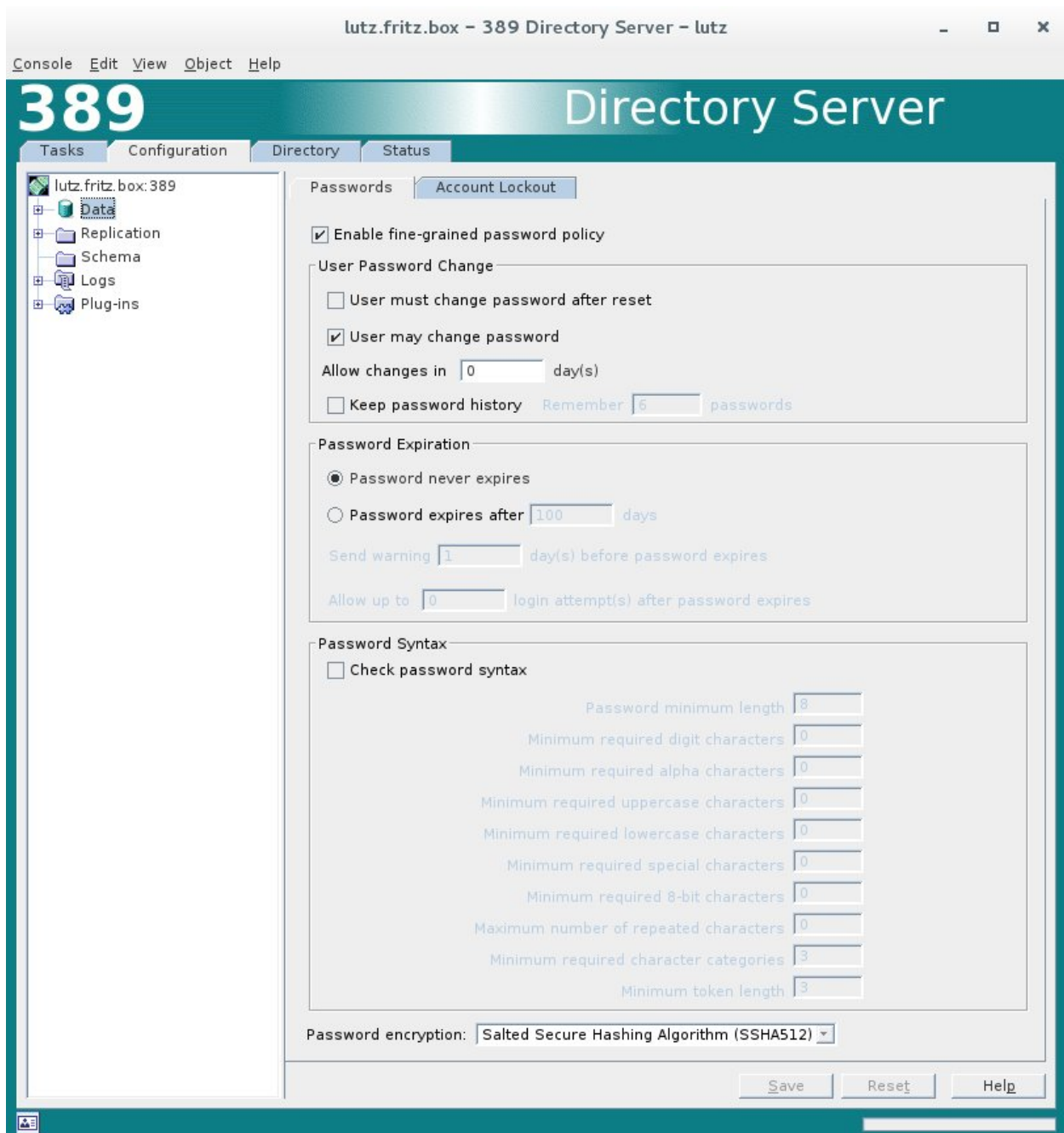


Fig. 2: LDAP Passwort Policy Einstellungen in 389-Console.

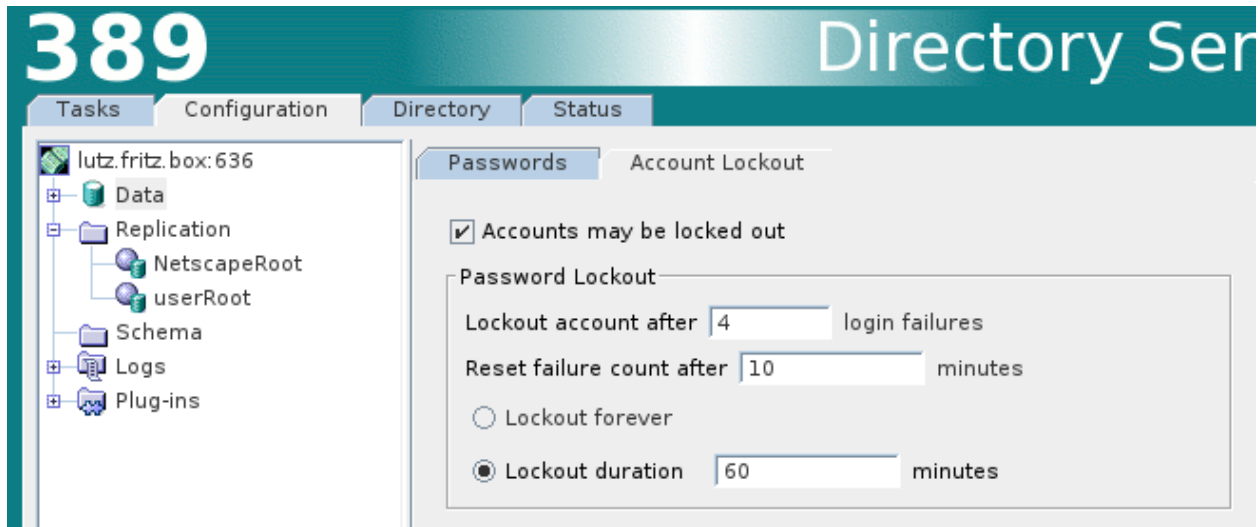


Fig. 3: LDAP Lockout Policy Einstellungen in der 389-Console.

db2bak.pl Dieses Skript ermöglicht Backups während des laufenden Servers zu erzeugen. Die Backups müssen später auch wieder bei einem laufenden Server eingespielt werden. Dieses Verfahren soll bei einer Single-Master Replikation auf dem Master Knoten eingesetzt werden.

cn=backup, cn=task, cn=config Ein solches Backup wird direkt über den LDAP Server erzeugt indem ein sog. Task angelegt wird. Der LDAP Server löscht den Task wenn das Backup abgeschlossen ist. Diese Möglichkeit benötigt deswegen auch einen laufenden LDAP Server für Backups und Wiederherstellung.

Aufgrund eines **Bugs** ist es `sssd` derzeit nicht möglich auf einen Backup LDAP-Server umzuspringen. Aufgrund dessen ist die Entscheidung der Backuplösung auf `db2bak.pl` gefallen. Bislang war ich aber noch nicht in der Lage mittels `db2bak.pl` ein Backup erfolgreich zu erzeugen. Bei einer Analyse des Quellcodes bin ich aber auch darüber gestolpert, dass der Nutzerlogin (ldap Nutzer und Passwort) direkt an `ldapmodify` übergeben werden, sodass beide in der Prozessliste zu finden sind. Aber ich konnte auch herausfinden, dass `db2bak.pl` nichts anderes macht als einen `cn=backup, cn=task, cn=config` Eintrag anzulegen was man auch selbst erledigen kann. Es wird daher auf eine Eigenentwicklung hinauslaufen deren Anforderungen wie folgt sind:

- Ein neuer Backup-Eintrag muss im LDAP erzeugt werden.
- Passwörter gehören nicht in die Prozessliste
- Die Verbindung mit dem Server soll mit der Root-CA abgeglichen werden.

1.3.5 Backup Wiederherstellung

1.4 Replikations Setup

Es gibt mehrere Arten von Replikationen die für das LDAP eingesetzt werden könne. Darunter bietet 389-Server die folgenden Optionen:

Single-Master Replikation Bei diesem Verfahren gibt es nur einen schreibfähigen Masterknoten dessen Informationen auf die sogenannten *Consumer* verteilt werden. Der Masterknoten wird als *Supplier* bezeichnet.

Multi-Master Replikation Die Multi-Master Replikation bietet die Möglich gleich mehrere Master Knoten (*Supplier*) aufzusetzen. Es ist maximal möglich 20 Supplier parallel aufzusetzen.

Cascading Replication Eine weitere Technik die jedoch nicht für das kleine Setup als Relevant betrachtet wird und daher nicht hier behandelt wird.

Um einen *Consumer* Server aufzusetzen ist es notwendig einen *Bind DN Entry* auf dem Consumer zu erzeugen. Mit einem solchen Eintrag bindet man einen Consumer an einen Supplier an. Anders ausgedrückt: Um mehrere LDAP-Datenbanken miteinander zu verknüpfen muss ein Eintrag auf den LDAP-Servern (Client/Consumer) die ihre Informationen von einem anderen Server (Server/Supplier) erhalten erzeugt werden. Ein solcher Eintrag muss auf jedem Consumer erzeugt werden und darf aus Sicherheitsgründen kein Teil der replizierten Datenbank sein (muss also z.B. in einem anderen Zweig gespeichert sein). Dieser Eintrag muss die folgenden Kriterien erfüllen:

- Einzigartigkeit
- Muss auf dem Consumer bestehen.
- Muss mit einem Eintrag auf dem Consumer Server in Verbindung stehen.
- Muss auf jedem Server erzeugt werden der Updates erhält.
- Darf kein Teil der replizierten Datenbank sein.
- Muss in der Replikationsvereinbarung auf dem Supplier definiert sein.
- Muss initial eine hohe idle timeout Zeitspanne haben um bei größeren Datenbanken die Replikation zu ermöglichen. Das kann mit dem Optionalen `nsIdleTimeOut` Attribut erreicht werden.

Um im Falle eines Totalausfalles immernoch flexible zu sein und auf beiden Rechner noch Einträge erzeugen zu können wird im folgenden das Setup für eine Multi-Master Replikation erläutert. Dazu ist es zunächst notwendig auf zwei Rechner bereits ein vollständiges LDAP-Setup zu haben. Das heißt das oben beschriebene Basis-Setup muss auf beiden Master Knoten durchgeführt werden.

Anschließend muss auf beiden Servern die Read-Write-Replikation aktiviert werden indem die Changelogs aktiviert werden, die zur Synchronisation eingesetzt werden. Siehe dazu [RW Replikationseinstellung für Masterknoten..](#)

Im folgenden wird der *Supplier Bind DN Entry* auf beiden Master Knoten eingerichtet, da beide Knoten jeweils für den anderen als Master fungieren. Dazu muss zunächst der LDAP Server gestoppt werden:

```
# systemctl stop dirsrv.target
```

Und im Anschluss die Datei `/etc/dirsrv/slapd-<instance>/dse.ldif` editiert werden. In dieser Datei wird ein neuer Benutzer zum Schluss der Datei eingeführt über den die Replikation durchgeführt werden soll. Das Passwort dieses Nutzers sollte dabei aus mindestens 25 Zeichen bestehen. Der Wert des `nsIdleTimeOut` Attributes wird in Sekunden angegeben und sollte initial so groß sein, dass eine vollständige Synchronisation möglich ist. Leider spezifiziert die Dokumentation nicht, ob der Wert 0 dabei den Timeout vollständig deaktiviert und wie man gute Werte ermitteln kann.

```
2736a2737,2745
> dn: cn=replication manager,cn=config
> objectClass: inetorgperson
> objectClass: person
> objectClass: top
> cn: replication manager
> sn: RM
> userPassword: mypassword
> passwordExpirationTime: 20380119031407Z
> nsIdleTimeout: 0
```

Anschließend können die beiden Master Knoten wieder gestartet werden:

```
# systemctl start dirsrv.target
```

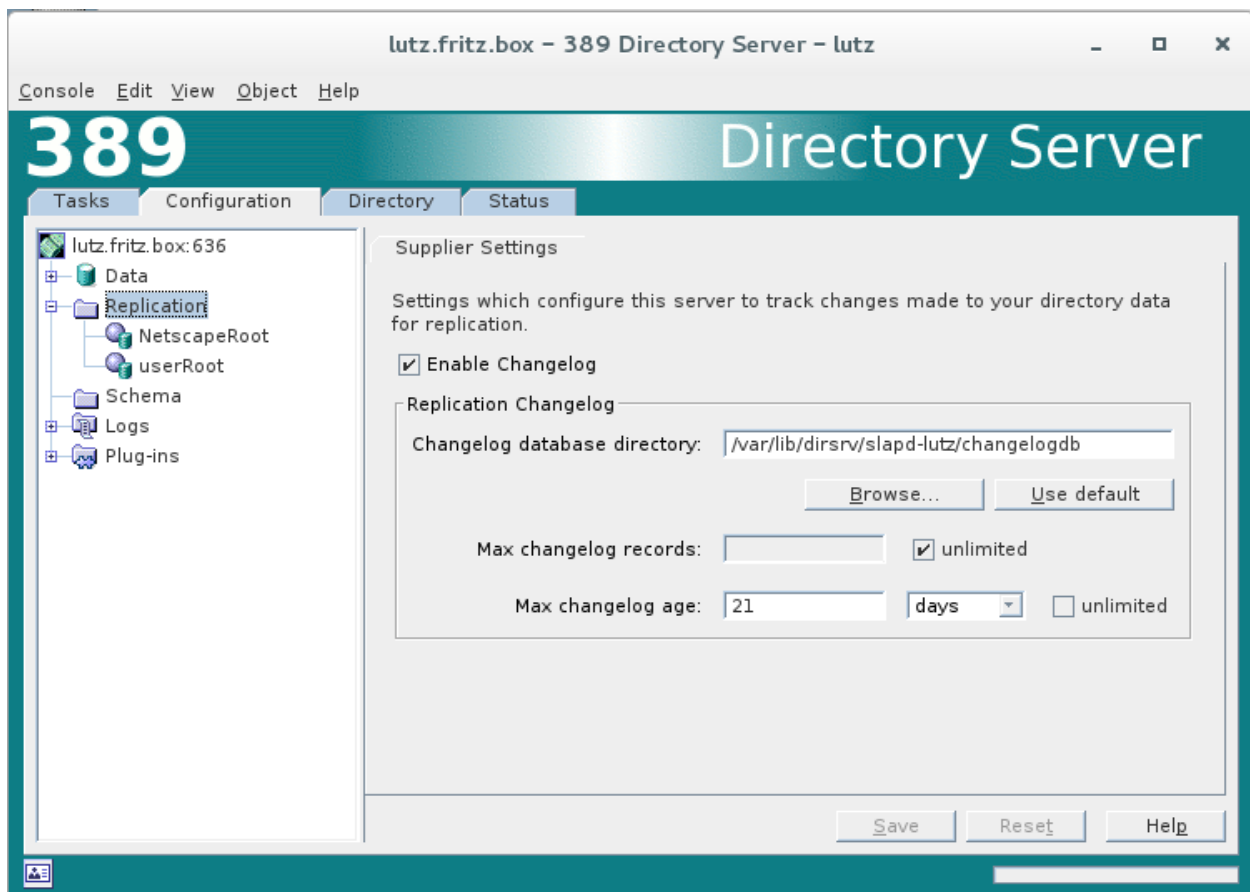


Fig. 4: RW Replikationseinstellung für Masterknoten.

Jetzt wird jeder LDAP-Server über die 389-Console so konfiguriert, dass die Replikation aktiviert wird. Dazu wählt man in der Console wieder im **Configuration** Tab die Replikation aus. Genauer soll der Unterpfad `userRoot` der alle Nutzerdaten enthält repliziert werden, weshalb die Einstellungen hierfür vorgenommen werden sollen. Für die Details siehe [Aktivierung der Replikation in der 389-Console](#), als **Replica ID** kann ein Wert zwischen 1 und 65534 frei gewählt werden z.B. das letzte IP Segment im VPN.

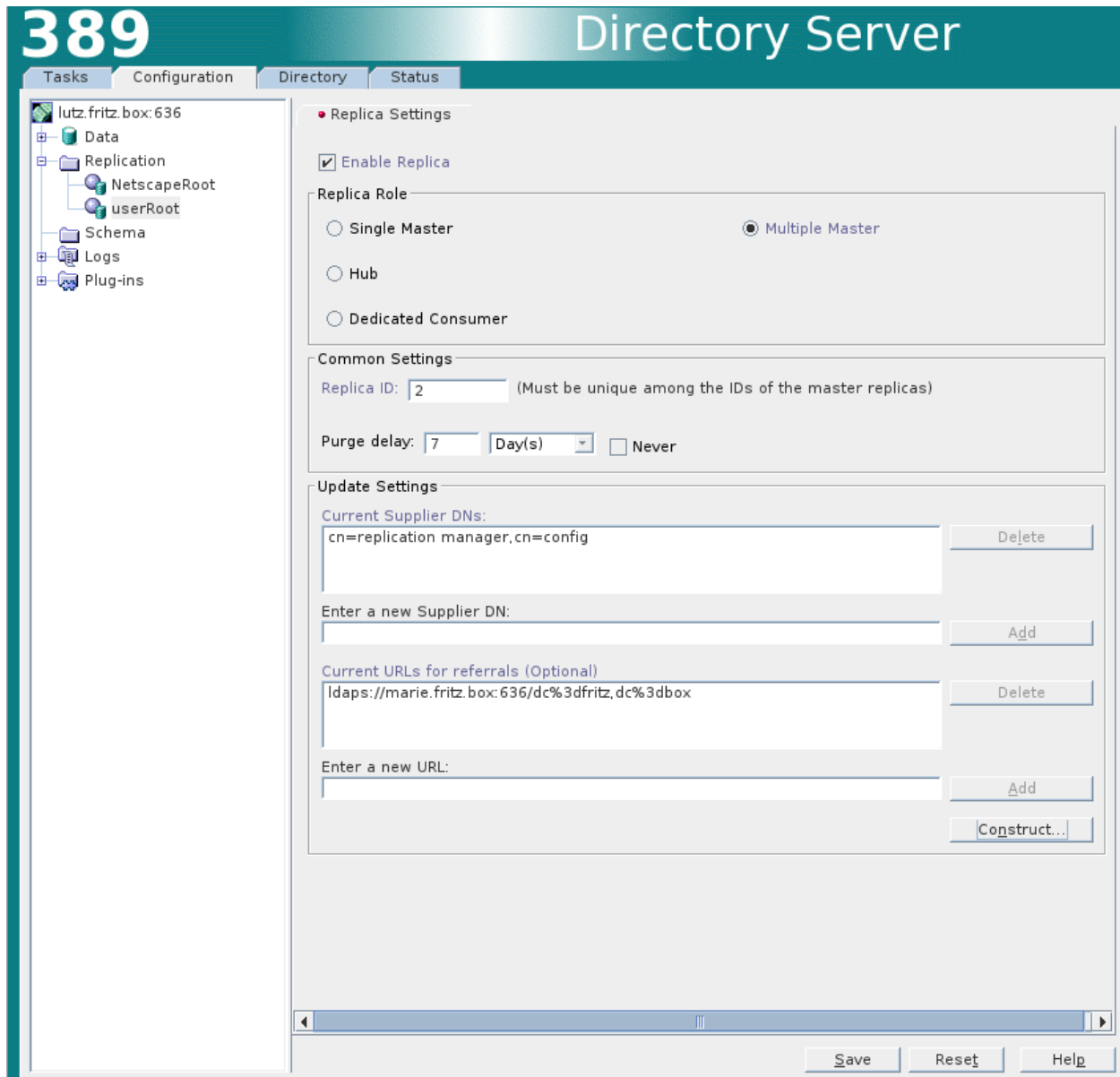


Fig. 5: Aktivierung der Replikation in der 389-Console

Im nächsten Schritt muss eine neue Replikationsvereinbarung erzeugt werden (siehe [Start des Wizards für eine neue Replikationsvereinbarung](#)).

Der Name und die Beschreibung könne frei gewählt werden dafür. Anschließend kann die Replikationsvereinbarung spezifiziert werden wie in [Start des Wizards für eine neue Replikationsvereinbarung](#) vorgestellt. Dabei ist es möglich, dass der jeweils andere Server als **Consumer** noch eingetragen werden muss. In der Folge kann man **Fractional Replication** konfigurieren die **nicht** in diesem Setup verwendet wird.

Da es innerhalb des LDAP nicht all zu häufig zu Änderungen kommen wird, sollen die Server immer in sync gehalten

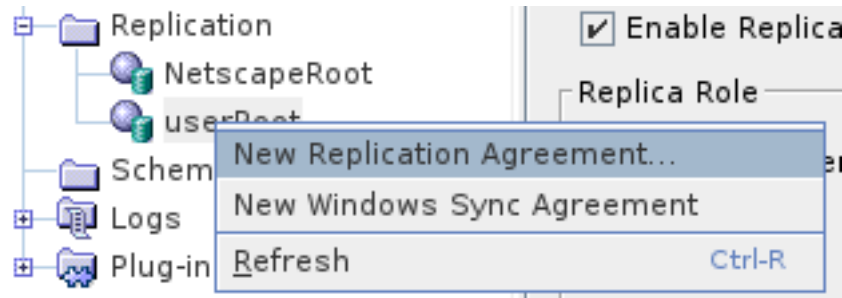


Fig. 6: Start des Wizards für eine neue Replikationsvereinbarung

werden. Das sollte (nein ich bin nicht sicher) bedeuten, dass ein Neustart der Server ohne Probleme immer durchführbar ist ohne die Agreements zu deaktivieren wenn nicht gerade ein Datensatz verändert wurde. Als Sicherungsschicht dienen auf alle Fälle die Backups. Zum Abschluss wird man gefragt ob man die consumer initialisieren soll, das darf **nur beim ersten mal** durchgeführt werden. Beim einrichten der weiteren Master-Server sollte man **Do not initialize consumer** wählen.

Zum Abschluss wurde noch die Synchronisation der Passwort-Policy aktiviert mittels:

```
root@shelob:~ # vim /tmp/globalpassword.ldif
dn: cn=config
changetype: modify
replace: passwordIsGlobalPolicy
passwordIsGlobalPolicy: on

root@shelob:~ # ldapmodify -x -D "cn=Directory Manager" -W -f /tmp/globalpassword.ldif
```

Das war erfolgreich, wenn im journal keine Logs mehr zu dem Thema geschrieben werden.

1.5 LDAP Client Anbindung

Das Programm `sssd` wird allgemein als Ersatz für die Kombination von `nss_ldap`, `pam_ldap` und `nslcd`. Die folgenden Man Pages werden von Red Hat zu dem Thema `sssd` als Referenzen empfohlen:

Funktionsbereich	
Allgemeine Konfiguration	<code>sssd.conf(5)</code>
sudo Services	<code>sssd-sudo</code>
LDAP Domains	<code>sssd-ldap</code>
Active Directory Domains	<code>sssd-ad sssd-ldap</code>
Identity Management (IdM oder IPA) Domains	<code>sssd-ipa sssd-ldap</code>
Kerberos Authentifizierung für Domains	<code>sssd-krb5</code>
OpenSSH Keys	<code>sss_ssh_authorizedkeys sss_ssh_knownhostsproxy</code>
Cache Einrichtung	<code>sss_cache (cleanup) sss_useradd, ...</code>

```
# yum install sssd-ldap
# vi /etc/sss/sss.conf
[sss]
domains = LDAP
services = nss, pam
config_file_version = 2
```


(continues on next page)


marie.fritz.box - 389 Directory Server - marie

ject Help

Source and Destination

Provide server and content information:

Supplier
 marie.fritz.box:389

Consumer
 lutz.fritz.box:636 Other...

Connection

☐ Use LDAP (no encryption)
☒ Use TLS/SSL (TLS/SSL encryption with LDAPS)
☐ Use StartTLS (TLS/SSL encryption with LDAP)

Authentication mechanism:

☐ Server TLS/SSL Certificate (requires TLS/SSL server set up)
☐ SASL/GSSAPI (requires server Kerberos keytab)
☐ SASL/DIGEST-MD5 (SASL user id and password)
☒ Simple (Bind DN/Password)

Bind as:

Password:

Subtree:
dc=fritz,dc=box

Back Next Cancel Help

Fig. 7: Start des Wizards für eine neue Replikationsvereinbarung

(continued from previous page)

```
[nss]
filter_groups = root
filter_users = root

[domain/LDAP]
enumerate = true
cache_credentials = TRUE

id_provider = ldap
auth_provider = ldap
ldap_schema = rfc2307
chpass_provider = ldap

ldap_uri = ldaps://shelob.lakrahn.de
ldap_search_base = dc=lakrahn,dc=de
ldap_tls_reqcert = demand
ldap_tls_cacert = /etc/sssdc/ca.crt

# chmod 600 /etc/sssdc/sssdc.conf
# authconfig --enablesssdc --enablesssdc_auth --update
```


CHAPTER 2

Indices and tables

- `genindex`
- `modindex`
- `search`