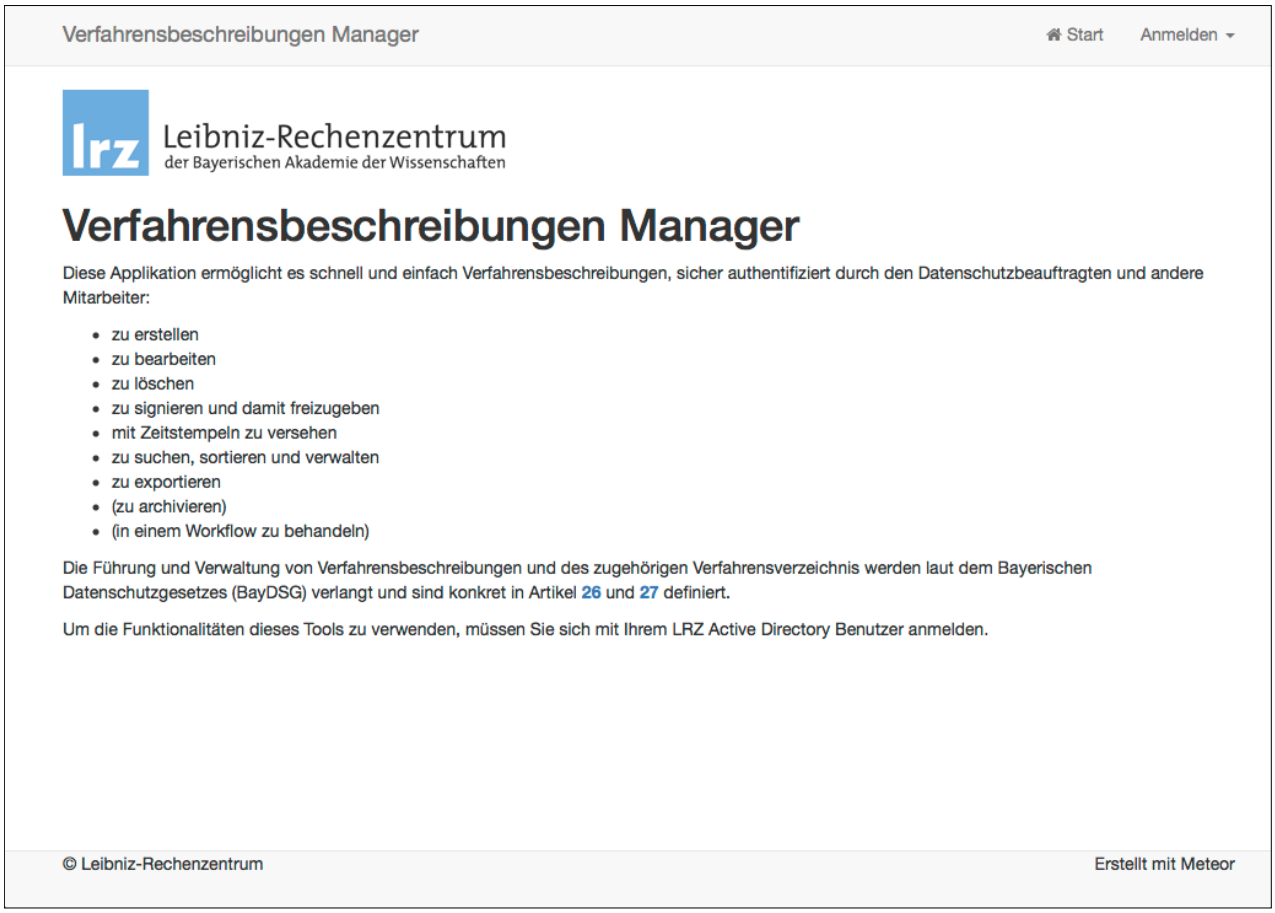

Procedure Description Manager

INSTALLATION MANUAL

Author: Steiert, Daniel

Created on: 8. Mai 2016

KURZFASSUNG

With this installation manual, it should be possible to set up the application on a fresh Debian 8 Linux machine.

It is a very technical and simple manual, with step-by-step instructions that basically just need to be copied and perhaps modified to fit the individual server setup.

All information written in italics are commands that can just be copied as is and might need to be changed slightly to fit the system-specific configuration.

Prerequisites

Create User:

adduser dstool

Update package manager and install dependencies:

apt-get update
apt-get upgrade
apt-get install curl g++ gcc make # to build node
*apt-get install build-essential g++ flex bison gperf ruby perl *
*libsqlite3-dev libfontconfig1-dev libicu-dev libfreetype6 libssl-dev *
libpng-dev libjpeg-dev python libx11-dev libxext-dev # to build webshot/phantomjs
apt-get install rsync # for syncing files

Install node:

su dstool
cd
Use any text-editor to create the file node-install.sh with the contents:

```
echo „echo 'export PATH=$HOME/local/bin:$PATH' >> ~/.bashrc
. ~/.bashrc
mkdir ~/local
mkdir ~/node-latest-install
cd ~/node-latest-install
curl https://nodejs.org/dist/v0.10.40/node-v0.10.40.tar.gz | tar xz —strip-components=1
./configure —prefix=~/local
make install # ok, fine, this step probably takes more than 30 seconds…
curl https://www.npmjs.com/install.sh | sh
```

chmod +x node-install.sh

./node-install.sh

curl https://install.meteor.com/ | sh
echo `export PATH=$HOME/.meteor:$PATH' >> ~/.bashrc

exit

Create service:

Use any text-editor to create the file/etc/systemd/system/dstool.service with the contents:

```
[Service]
ExecStart=/home/dstool/local/bin/node /var/dstool/bundle/main.js
Restart=always
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=dstool
User=dstool
Group=dstool
Environment=NODE_ENV=production
Environment=MONGO_URL=mongodb://localhost/meteor
Environment=ROOT_URL=https://dstool.srv.lrz.de
Environment=PORT=4000
Environment=PATH=/usr/bin
Environment=NODE_TLS_REJECT_UNAUTHORIZED=0 // for ldap
[Install]
WantedBy=multi-user.target
```

mkdir /var/dstool
chown dstool:dstool /var/dstool
systemctl enable dstool

Install MongoDb:

sudo apt-get install mongodb

Create Service for MongoDb:

Use any text-editor to create the file/etc/systemd/system/mongodb.service with the contents:

```
[Unit]
Description=NoSQL database
After=network.target
[Service]
ExecStart=/usr/bin/mongod --quiet --config /etc/mongodb.conf
Restart=always
StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=mongodb
User=mongodb
Group=mongodb
Environment=LC_ALL=C
[Install]
WantedBy=multi-user.target
```

Enable Service for MongoDb:

systemctl enable mongodb

Setup Apache2 Proxy:

apt-get install apache2
a2enmod rewrite
a2enmod proxy
a2enmod proxy_http
a2enmod proxy_wstunnel
a2enmod ssl

Create TLS certificate:

create ssl dir
mkdir /etc/apache2/ssl
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/dstool.key -out /etc/apache2/ssl/dstool.crt
chmod -R 600 /etc/apache2/ssl

Use any text-editor to create the file/etc/apache2/sites-available/dstool.conf with the contents:

```
Define meteorport "4000"

<VirtualHost *:80>
        ServerAdmin webmaster@12steiert.de

        ServerName dstool.srv.lrz.de

        RewriteEngine on
        ReWriteCond %{SERVER_PORT} !^443$
        # This allows DDP clients like ObjectiveDDP and Meteor-Unity to connect
        RewriteRule ^/websocket wss://%{HTTP_HOST}/websocket [NC,R,L]
        # This allows the meteor webapp to connect
        RewriteRule ^/sockjs/(.*)/websocket wss://%{HTTP_HOST}/sockjs/$1/websocket [NC,R,L]
        RewriteRule ^/(.*) https://%{HTTP_HOST}/$1 [NC,R,L]

</VirtualHost>
<VirtualHost *:443>
    ServerAdmin webmaster@12steiert.de
    ServerName https://dstool.srv.lrz.de

    SSLEngine on

    SSLProxyEngine On
    ProxyRequests Off # Disable forward proxying

    # This allows DDP clients like ObjectiveDDP and Meteor-Unity to connect
    ProxyPass /websocket ws://localhost:${meteorport}/websocket

    # This allows the meteor webapp to connect
    ProxyPassMatch ^/sockjs/(.*)/websocket ws://localhost:${meteorport}/sockjs/$1/websocket
```

```
    ProxyPass / http://localhost:${meteorport}/
    ProxyPassReverse / http://localhost:${meteorport}/

    SSLCertificateFile    /etc/apache2/ssl/dstool.crt
    SSLCertificateKeyFile /etc/apache2/ssl/dstool.key

    BrowserMatch "MSIE [2-6]" \
       nokeepalive ssl-unclean-shutdown \
       downgrade-1.0 force-response-1.0
    # MSIE 7 and newer should be able to use keepalive
    BrowserMatch "MSIE [17-9]" ssl-unclean-shutdown

</VirtualHost>
```

a2ensite dstool
service apache2 restart

apt-get install ldap-utils

taken from https://ldapwiki.com/wiki/Obtain%20a%20Certificate%20from%20Server#section-Obtain+a+Certificate+from+Server-UsingLdapsearch
ldapsearch -x -T ~/ -t -h your-edirectory-host.yourdomain.com -b "cn=Security" objectclass=nDSPKICertificateAuthority cACertificate
openssl x509 -inform der -in ~/ldapsearch-cACertificate-FS7uCC -out ~/ldapserver.pem

add ldapserver.pem to project and set up for ssl

add file to /etc/ssl/certs

c_rehash
update-ca-certificates

mkdir /var/archives
chown -R dstool:dstool /var/archives

Add process.env.ARCHIVE_PATH in config file /etc/systemd/system/dstool.service

Copy all project files to the dstool user's home-directory in a folder named pdmanager-src

In /home/dstool/pdmanager-src execute:

./deploy-server

The server should then be built on the machine and deployed to the folder /var/dstool before restarting the service.