SHEBANQ Installation on Ubuntu 12.04 LTS (public)

IP and DOMAIN

Allocate a domain from a domain hoster and connect it to the IP address of your server.

Set the reverse lookup for your server to this domain.
For this manual we assume we are working with shebanq.ancient-data.org .

Check by doing ssh from command line.

```
ssh root@shebanq.ancient-data.org
```

FIREWALL
We need port 80, 443, 8080, 22

USER ACCOUNTS, SUDOer

We have only one user, say dirkr who is a sudoer.
Root login will be disabled.

```
addgroup admin
adduser dirkr
adduser dirkr admin
```

Take password from your password manager:

DISABLE root login

```
passwd -l root
```

From now on: be logged in as dirkr, and then: sudo -i .

INSTALL CONVENIENT PACKAGES

Bring all installed packages up to date.

```
apt-get update
apt-get upgrade
```

Install additional packages

```
apt-get install git tomcat6 mysql-server bzip2 man-db vim python-pip python-lxml python-
dev python-software-properties curl make g++ zlib1g-dev binutils swig autoconf automake
libtool
libwxgtk2.8-0 libwxgtk2.8-dev openjdk-6-jdk libmysqlclient-dev maven
```

At some point dpkg asks for the mysql root password. Use one from your password manager.

Install additional python packages

```
pip install gitpython
pip install purl
pip install requests
```

SSH Keys

We work with an ssh certificate and disable login with user name and password.
So, logging in is very easy (no password dialog) and very secure.
However, in order to do business on the server, a sudo -i is almost alway needed, and then the password for dirkr is still needed.

```
mkdir /home/dirkr/.ssh
```

On your client machine (mac or linux), generate a key (use a difficult passphrase)

```
ssh-keygen -t rsa
cat ~/.ssh/id_rsa.pub | ssh dirkr@shebanq.ancient-data.org "cat >> ~/.ssh/
authorized_keys"
```

On the server again:

```
sudo -i
```

Verify in the ssh config file the following fragments:

```
vim /etc/ssh/sshd_config

~~~~~~~~
PermitRootLogin no
Protocol 2
PubkeyAuthentication yes
AuthorizedKeysFile     %h/.ssh/authorized_keys
~~
Begin Passwords and PAM related
PasswordAuthentication no
ChallengeResponseAuthentication no
~~
```

It seems that the following is notvery important whether yes or no, as long as the previous twp are no.

```
~~
UsePAM yes
End Passwords and PAM related
AllowUsers dirkr
~~~~~~~
```

Then check, first

```
service ssh restart
```

Keep this connection open when checking in another tab of the terminal.
Check this by logging out and getting back in.

DROPBOX (for backup)

We use dropbox for a simple backup.
There is only one mysql database that changes, the shebanq database.
Every day at 19.00 this database will be dumped, gzipped and put into the dropbox.
Also, all installation packages and data that cannot readily be downloaded is stored in the folder installation-sources.
And a few config files that need to be tweaked are stored in their tweaked form (for shebanq, shemdros, apache).

Download Dropbox

```
cd /root
wget -O - http://www.dropbox.com/download?plat=lnx.x86_64 | tar xzf -
```

Install Dropbox

```
.dropbox-dist/dropboxd
```

If first time, copy the https link from terminal to browser (on arbitrary machine).
Log in into dropbox with your dropbox account first.
Wait on console for welcome message.
Kill the program by Ctrl-C.

If later on you change the password of the dropbox account, this dropbox client stays connected to the dropbox server. No action here needed!

Add dropbox script to /etc/init.d (with name dropbox and with following text):

```
vim /etc/init.d/dropbox

=====
start() {
    echo "Starting dropbox ..."
    start-stop-daemon -b -o -c root -S -x /root/.dropbox-dist/dropboxd
}
stop() {
    echo "Stopping dropbox ..."
    start-stop-daemon -o -c root -K -n dropbox
}
status() {
        dbpid=$(pgrep -u root dropbox)
        if [ -z $dbpid ] ; then
            echo "dropbox not running."
        else
            echo "dropbox running."
        fi
}
case "$1" in
  start)
    start
    ;;
  stop)
    stop
    ;;
  restart|reload|force-reload)
    stop
    start
    ;;
  status)
    status
    ;;
  *)
    echo "Usage: /etc/init.d/dropbox {start|stop|reload|force-reload|restart|status}"

    exit 1

esac
exit 0

=====
```

Set permissions

```
chmod u+x /etc/init.d/dropbox
```

Check

```
/etc/init.d/dropbox status
/etc/init.d/dropbox start
/etc/init.d/dropbox stop
```

Once the dropbox is running, it will download content to the server, especially the tweaked config files and the installation files and the last dump of the shebanq database.

Set dropbox to start at boot

```
update-rc.d dropbox defaults
```

Check whether this works by rebooting.

```
/etc/init.d/dropbox status
```

gives

```
dropbox running.
```

(In order to remove:
Move the script to the roots home directory (so it is easy to enable the service later on)
```
mv /etc/init.d/dropbox ~
update-rc.d dropbox remove
```
)

Make a subfolder in the dropbox folder
(or let the dropbox folder be filled with pre-existing material)

This folder collects all the backup data:

(only if there is no content in the Dropbox to start with
```
mkdir Dropbox/shebanq.ancient-data.org
mkdir Dropbox/shebanq.ancient-data.org/mysql
```
)

Make a backup script

Create a backup script backup.sh to daily dump the mysql database and log the action in a specific log file.

```
cd ~
vim backup.sh

=====
#!/bin/sh

dest="/root/Dropbox/shebanq.ancient-data.org/mysql"
logdest="/var/log/mysqldump.log"

if [ ! -e $dest ]
then
        mkdir $dest
fi

echo -n MySQL dump at $(date) " ... " >> $logdest

mysqldump --defaults-extra-file=/root/mysqldumpopt shebanq | gzip > $dest/shebanq.sql.gz
chmod go-rwx $dest/shebanq.sql.gz

if [ $? != 0 ]
then
        echo "Wrong $(date)" >> $logdest
else
        echo "OK $(date)" >> $logdest
fi
========
```

Give it execute permission, but protect it:

```
chmod go-rwx backup.sh
chmod u+x backup.sh
```

Add the password in an optionfile

```
vim mysqldumpopt

====
[mysqldump]
password = 'mysql root password'
user = root
====
```

Protect it (because it contains the mysql rootpassword):
```
chmod go-rwx mysqldumpopt
```

Let the backup script run daily. Add the backup script to a root crontab.
See **maintenance** below.

Installation files

If not already on the dropbox, collect and save the following files in installation-sources

```
cd ~/Dropbox/shebanq.ancient-data.org
mkdir installation-sources
```

You should get the following content.

```
E -rw-r--r-- 1 dirkr dirkr 11139138 Dec 30 21:44 emdros-3.4.0.tar.gz
M -rw-r--r-- 1 dirkr dirkr 23360605 Dec 30 21:41 etcbc4.mql.bz2
P -rw-r--r-- 1 dirkr dirkr 10736978 Dec 31 01:01 etcbc4-passage.sql.bz2
W -rwxr-xr-x 1 root root       4294 Dec 31 00:12 setup-web2py-ubuntu.sh
A -rw-r--r-- 1 root root       2050 Jan  7 14:22 sites-available-default
SB-rw------- 1 root root        199 Jan  7 15:24 usr-local-shebanq_db.cfg
SD-rw-r--r-- 1 root root         75 Jan  7 15:25 usr-local-shemdros.cfg
R -rw-r--r-- 1 root root         85 Jan  7 22:35 web2py-routes.py
T drwxr-xr-x 3 dirkr dirkr     4096 Jan  7 23:26 shemdros
```

E = the emdros source file, as downloaded from http://emdros.org, more directly: http://sourceforge.net/projects/emdros/files/emdros/3.4.0/emdros-3.4.0.tar.gz/download

M = MQL dump of ETCBC database, as obtained by dumping the bhs4 database on the jakob server of the ETCBC institute, or by downloading is from DANS (https://easy.dans.knaw.nl/ui/datasets/id/easy-dataset:58245, goto datafiles, folder sourcedata, file etcb4.mql.bz2, 23,360,605 bytes).

P = SQL dump of the passage database, assembled by LAF-Fabric for SHEBANQ. See http://nbviewer.ipython.org/github/ETCBC/laf-fabric-nbs/blob/master/shebanq/laf2shebanq.ipynb

W = modified setup script for web2py. See http://web2py.com/books/default/chapter/29/13/deployment-recipes#One-step-production-deployment

A = modified apache default site. First the web2py setup script works and changes the apache default. But that is for Apache 2.4.0 on Ubuntu 14.04 and we still have 12.04. So replace that by the contents of this file. See below in the appendix.

SB= config file for shebanq. To be moved to /usr/local/shebanq_db.cfg. See below in the appendix.

SD= config file for shemdros. To be moved to /usr/local/shemdros.cfg. See below in the appendix.

R= config file for web2py, stating that shebanq is the default application. See below in the appendix.

T= binary files of shemdros, working for Java 1.6. Downloadable from: https://www.dropbox.com/sh/nhwuvghc3afui8j/AAB4wndZKeBicdE-gP8qsr16a?dl=0

Check whether the backup data shows up on the dropbox site.

Copy these files to convenient locations

```
cd ~
mkdir tmp
cp Dropbox/shebanq.ancient-data.org/installation-sources/* ~/tmp
cp ~/Dropbox/shebanq.ancient-data.org/mysql/shebanq.sql.gz ~/tmp
cd tmp
```

MYSQL
Installation of mysql is straightforward and has already been done by now.
Make sure mysql works with utf8.

```
vim vim /etc/mysql/my.cnf

[mysql]
```

```
#no-auto-rehash     # faster start of mysql but no tab completition
default-character-set=utf8

service mysql restart
```

MYSQL USERS AND DATABASES

There will the following databases:
etcbc4, which is the emdros Hebrew Text Database.
shebanq, which contains the data for the shebanq web app (users, saved queries, metadata).
passage, which contains the texts that the webapp has to show.

There will be the following users
shemdros acting on behalf of the shemdros service. It will have readonly rights for the **etcbc4** database.
shebanq acting on behalf of the shebanq web application.
It will have readonly rights for the **etcbc4** and **passage** databases,
and read/write access for the **shebanq** database.

```
mysql -u root -p

GRANT SELECT ON etcbc4.* TO shemdros@localhost IDENTIFIED BY 'mysql pwd for shemdros'
WITH GRANT OPTION;
GRANT SELECT ON passage.* TO shebanq@localhost IDENTIFIED BY 'mysql pwd for shebanq'
WITH GRANT OPTION;
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER ON shebanq.* TO shebanq@localhost
IDENTIFIED BY 'mysql pwd for shebanq' WITH GRANT OPTION;
FLUSH PRIVILEGES;
```

Checking:

```
SELECT User, Host, Password FROM mysql.user;
SHOW GRANTS FOR 'etcbc4'@'localhost';
SHOW GRANTS FOR 'shemdros'@'localhost';
SHOW GRANTS FOR 'shebanq'@'localhost';

exit
```

EMDROS

```
cd ~/tmp
```

Unpack in an arbitrary directory

```
tar xvf emdros-3.4.0.tar.gz
```

Configure for installation in /opt

```
cd emdros-3.4.0
./configure --prefix=/opt/emdros --with-sqlite3=local --with-mysql=yes --with-swig-
language-java=yes --with-swig-language-python=yes --with-jdk- dir=/usr/lib/jvm/java-7-
openjdk-amd64
```

Build and install.

```
make
make install
```

Now mql is in /opt/emdros/bin/mql

For SHEMDROS we need the build directory for additional linking, so we move the built files to the installation location and rename it to emdros-src.

```
cp -r emdros-3.4.0 /opt/emdros-src
```

MYSQL DATA

Passages database. Is dumped in the file etcbc4-passage.sql

cd ~/tmp

```
bunzip2 etcbc4-passage.sql.bz2
mysql -u root -p <etcbc4-passage.sql

gunzip shebanq.sql.gz
mysql -u root -p
create database shebanq;
use shebanq;
source shebanq.sql
```

Test:

```
select first_name, last_name, email from auth_user;
```

which yields something like

```
| Daniel | Naumann | ...@... |
| Neat | Randriamialison | ...@... |
| Jesse | Griffin | ...@... |
| Austen | Du | ...@... |
| Nathan | Bierma | ...@... |
| Nicolai | Winther-Nielsen | ...@... |
| Lejeune | Jean | ...@... |
| Brigitte | MINEL | ...@... |
+------------+-----------------+----------------------------------+

use passage;
select count(*) from verse;
```

Which yields:

```
+----------+
| count(*) |
+----------+
|    23213 |
+----------+

exit
```

EMDROS DATA

The etcbc4 mql database can be loaded as follows.
If updating, drop the database first:

```
mysql -u root -p
drop database etcbc4;
exit

Then:

bunzip2 etcbc4.mql.bz2
/opt/emdros/bin/mql -b m -u root -p 'mysql pwd for root' <etcbc4.mql
```

EMDROS-SHEMDROS ADDITIONAL CONFIG
Setting up paths. Create a file with path statements as follows:

```
vim /etc/profile.d/emdros.sh

with contents
=====
EMDROS_HOME=/opt/emdros
export EMDROS_HOME
PATH=$EMDROS_HOME/bin:$PATH
export PATH
=====
```

```
chmod a+x /etc/profile.d/emdros.sh
```

Reboot. Then mql just works.
After log in:

```
which mql
          /opt/emdros/bin/mql
```

preparation for shemdros: java bindings

```
sudo -i

mkdir /usr/local/lib/emdros
cd /usr/local/lib/emdros/
ln -s /opt/emdros-src/SWIG/java/jemdros.jar jemdros.jar
ln -s /opt/emdros-src/SWIG/java/.libs/libjemdros.so libjemdros.so
ln -s /opt/emdros-src/harvest/.libs/libharvest.so libharvest.so

mkdir /opt/emdros-java
cd /opt/emdros-java
cp /opt/emdros-src/SWIG/java/TestEmdros.java .
javac -cp /usr/local/lib/emdros/jemdros.jar:. TestEmdros.java
java -cp /usr/local/lib/emdros/jemdros.jar:. TestEmdros
```

WEB2PY
Either just follow the script as in the installation sources, or download the script and edit it before running.

The short way:

```
cd ~/tmp
./setup-web2py-ubuntu.sh
```

The long way:

Follow the Web2Py 1-step guide, but skip the postgres steps
(see http://web2py.com/book/default/chapter/13#One-step-production-deployment)

```
cd /home/dirkr
wget http://web2py.googlecode.com/hg/scripts/setup-web2py-ubuntu.sh
chmod +x setup-web2py-ubuntu.sh

vim setup-web2py-ubuntu.sh
```

Comment the lines that install `postgresql python2.5 python2.5-psycopg2 apache2`
Comment the line that restarts `postgresql`

```
./setup-web2py-ubuntu.sh
```

Here the long and the short way converge again.

Questions:
Dialog Postfix Configuration: Internet Site

For the self-signed certificate, enter:

```
Country Name (2 letter code) [AU]:NL
State or Province Name (full name) [Some-State]:Zuid-Holland
Locality Name (eg, city) []:Den Haag
Organization Name (eg, company) [Internet Widgits Pty Ltd]:KNAW
Organizational Unit Name (eg, section) []:DANS
Common Name (e.g. server FQDN or YOUR name) []:shebanq.ancient-data.org
Email Address []:dirk.roorda@dans.knaw.nl
```

The script asks for admin password in the end. Take that from the password manager.

If you want to change that later, do the following.

(Problems with long 32 char password. Took an intial piece of 16 chars from a new password.)

```
cd /home/www-data/web2py
sudo -u www-data python -c "from gluon.widget import console; console();"
sudo -u www-data python -c "from gluon.main import save_password;
save_password(raw_input('admin password: '),443)"
```

End change password.

Finally save the web2py config for the apache default site:

```
cp ~/tmp/sites-available-default /etc/apache2/sites-available/default
```

Management

Session files build up quickly, the sessions directory has to be cleaned.
This can be done by running the following command (background, does cleaning every five minutes):

(I have not chosen this way

```
nohup python /home/www-data/web2py/web2py.py -S shebanq -M -R /home/www-data/web2py/
scripts/sessions2trash.py &
```

Take care that this happens on reboot, I do not yet know how best to do this in the best way
)

(I have chosen this. It will be done in the section management
Do this as an hourly cron job of this.
)

SHEBANQ

Here comes the actual shebanq web application! We clone it from github.
For updates, we pull from github. This can be done on the command line, but also remotely, on the web2py administrative app.

```
cd /home/www-data/web2py/applications
su www-data
git clone https://github.com/Dans-labs/shebanq
exit
```

(In case of updating:
either on the commandline:

```
cd /home/www-data/web2py/applications/shebanq
su www-data
git pull origin master
exit
service apache2 restart
```

or use the administrative application of web2py
end updating)

```
mkdir -p /usr/local/shemdros
mkdir /usr/local/shebanq
```

Copy the config files in place:

```
cp ~/tmp/usr-local-shebanq_db.cfg /usr/local/shebanq/shebanq_db.cfg
cp ~/tmp/usr-local-shemdros.cfg /usr/local/shemdros/shemdros.cfg
cp ~/tmp/web2py-routes.py /home/www-data/web2py/routes.py
```

Restore the databases files (just metadata about tables that web2py somehow needs).
If you start with a new shebanq database, this is not needed!

```
mkdir /home/www-data/web2py/applications/shebanq/databases
cp /root/Dropbox/shebanq.ancient-data.org/shebanq-databases/* /home/www-data/web2py/
applications/shebanq/databases
```

```
chown -R www-data:www-data /home/www-data/web2py/applications/shebanq/databases

service apache2 restart
```

Check it out:
http://shebanq.ancient-data.org the shebanq site
https://shebanq.ancient-data.org/shebanq/appadmin shebanq maintenance
https://shebanq.ancient-data.org/admin web2py admin app, use the web2py admin password

SHEMDROS

This is the webservice that can execute MQL queries against the etcbc4 database and deliver results.
It is called via a rest-api by the shebanq web application.
It is a tomcat war, basically.

From source (does not work yet)
Make this working with HenkB

Download the sources from Github: https://github.com/Dans-labs/shemdros,
or say

```
cd ~/tmp
git clone https://github.com/Dans-labs/shemdros
```

Then install it by

```
cd shemdros
mvn clean install -DskipTests -Dmaven.javadoc.skip=true
```

End (from source)

BINARY

```
cp -r ~/tmp/shemdros /opt
chown -R tomcat6:tomcat6 shemdros
```

(re)start shemdros (needed when etcbc4 database has been reimported)

```
/opt/shemdros/shemdros install
```

MAINTENANCE

SHEBANQ:
Database backup every night at 01 past 19 o'clock.
Session cleaning every hour.

SHEMDROS
Restart tomcat6 every 10 minutes, because of faulty, long running and orphaned MQL queries.

Do all these tasks as cron jobs:

```
crontab -e

01 19 * * * /root/backup.sh
00 * * * * python /home/www-data/web2py/web2py.py -Q -S shebanq -M -R /home/www-data/
web2py/scripts/sessions2trash.py -A -o
00 * * * * service tomcat6 restart
10 * * * * service tomcat6 restart
20 * * * * service tomcat6 restart
30 * * * * service tomcat6 restart
40 * * * * service tomcat6 restart
50 * * * * service tomcat6 restart
```

See /var/log/syslog to see if the cron jobs run without errors.

DROPBOX Additional backups

Add material to the dropbox

```
cd ~/Dropbox/shebanq.ancient-data.org
```

(databases dir in shebanq web2py app, if not already present:

```
mkdir shebanq-databases
cp /home/www-data/web2py/applications/shebanq/databases/* shebanq-databases
```
)

Apache config file (copy)
```
cp /etc/apache2/sites-available/default sites-available-default
```

All config files (linked to dropbox)
```
ln -sfv /etc etc
```

Before testing the backup script, copy a previous export of the database out of the way:

```
cd Dropbox/shebanq.ancient-data.org/mysql
cp shebanq.sql.gz shebanq-previous.sql.gz
```

Test it:

```
./backup.sh
ls -lh Dropbox/demo.datanetworkservice.nl/mysql
```

gives

-rw------- 1 root root 19M 2012-04-27 16:14 mysql.sql.gz

APPENDIX A contents of Apache default site

```
NameVirtualHost *:80
<VirtualHost *:80>
    ServerName shemdros.ancient-data.org
    ProxyPass / http://localhost:8080/shemdros/
    ProxyPassReverse / http://localhost:8080/shemdros/
</VirtualHost>

<VirtualHost *:80>
  ServerName shebanq.ancient-data.org

  #RewriteEngine On
  #RewriteCond %{HTTPS} !=on
  #RewriteRule ^/?(.*) https://%{SERVER_NAME}/$1 [R,L]

  WSGIDaemonProcess web2py user=www-data group=www-data
  #processes=1 threads=1
  WSGIProcessGroup web2py
  WSGIScriptAlias / /home/www-data/web2py/wsgihandler.py
  WSGIPassAuthorization On

  <Directory /home/www-data/web2py>
    AllowOverride None
    Order Allow,Deny
    Deny from all
    <Files wsgihandler.py>
      Allow from all
    </Files>
  </Directory>

  AliasMatch ^/([^/]+)/static/(?:_[\d]+.[\d]+.[\d]+/)?(.*) \
        /home/www-data/web2py/applications/$1/static/$2
```

```
  <Directory /home/www-data/web2py/applications/*/static/>
    Order Allow,Deny
    Allow from all
    Options -Indexes
    ExpiresActive On
    ExpiresDefault "access plus 1 hour"
  </Directory>

  <Location /admin>
    Deny from all
  </Location>

  CustomLog /var/log/apache2/access.log common
  ErrorLog /var/log/apache2/error.log
</VirtualHost>

<VirtualHost *:443>
  SSLEngine on
  SSLCertificateFile /etc/apache2/ssl/self_signed.cert
  SSLCertificateKeyFile /etc/apache2/ssl/self_signed.key

  WSGIDaemonProcess web2pys user=www-data group=www-data
  # processes=1 threads=1
  WSGIProcessGroup web2pys
  WSGIScriptAlias / /home/www-data/web2py/wsgihandler.py
  WSGIPassAuthorization On

  <Directory /home/www-data/web2py>
    AllowOverride None
    Order Allow,Deny
    Deny from all
    <Files wsgihandler.py>
      Allow from all
    </Files>
  </Directory>

  AliasMatch ^/([^/]+)/static/(?:_[\d]+.[\d]+.[\d]+/)?(.*) \
        /home/www-data/web2py/applications/$1/static/$2

  <Directory /home/www-data/web2py/applications/*/static/>
    Order Allow,Deny
    Allow from all
    Options -Indexes
    ExpiresActive On
    ExpiresDefault "access plus 1 hour"
  </Directory>

  CustomLog /var/log/apache2/ssl-access.log common
  ErrorLog /var/log/apache2/error.log
</VirtualHost>
```

APPENDIX B config file for shebanq

```
# /usr/local/shebanq/shebanq_db.cfg
[passage]
host = localhost
user = shebanq
passwd = mysql password for shebanq user
db = passage

[shebanq]
host = localhost
user = shebanq
passwd = mysql password for shebanq user
db = shebanq
```

APPENDIX C config file for shemdros

```
[server]
scheme = http
host = shemdros.ancient-data.org
port = 80
root = /
```

APPENDIX D config file for web2py

```
routers = dict(
    BASE = dict(
        default_application='shebanq',
    )
)
```