SHEBANQ Installation on Ubuntu 12.04 LTS (public)

We restrict ourselves to the application specific details. For general systems management info, we recommend the Ubuntu Server Guide for 12.04 LTS.

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 .
If you are setting up a shebanq outside this domain, take the appropriate actions.

FIREWALL
We need port 80 (http), 443 (https), 22 (ssh)

USER ACCOUNTS, SUDOer
Most things need to be installed with root access, either by root or by a sudo-er. We assume that the server can be accessed by console or ssh.

INSTALL CONVENIENT PACKAGES

Bring all installed packages up to date.

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

Install additional packages
At some point dpkg will ask for the mysql root password.
Generate and store a strong password.
In the sequel, we refer to it as pwd-mysql-root.

```
apt-get install git 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
libmysqlclient-dev

pip install gitpython
pip install markdown
```

Installation files
Here is a SURFdrive to a folder with the installation files.
Get them into a handy location on the server, say `/root/tmp`.

You should get the following content.

```
E   emdros-3.4.0.tar.gz
M   shebanq_etcbcx.mql.bz2   (for x = 4, 4b, ...)
P   shebanq_passagex.sql.bz2 (for x = 4, 4b, ...)
W   setup-web2py-ubuntu.sh
A   sites-available-default
R   web2py-routes.py
```

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). There may be several versions, they all need to be installed.

P = SQL dump of the passage database, assembled by LAF-Fabric for SHEBANQ. See https://shebanq.ancient-data.org/shebanq/static/docs/tools/shebanq/laf2shebanq.ipynb. There may be several versions, they all need to be installed.

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.

R= config file for web2py, declaring shebanq as the default application. See below in the appendix.

```
cd ~/shebanq-install
```

MYSQL CONFIG

Make sure mysql works with utf8.
Set the default-character-set to 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: (for x = 4 or 4b or 4s, or ... depending on what is present)
shebanq_etcbc*x*, which is the emdros Hebrew Text Database version ***x***,
shebanq_passage*x*, which contains the texts that the webapp has to show;

There are two databases with content added by the users.
shebanq_web, which contains the data for the shebanq web app (users, saved queries, metadata).
shebanq_note, which contains the manual notes added by the users.

There will be the following user:
shebanq acting on behalf of the shebanq web application.
It will have readonly rights for the **shebanq_etcbc*x*** and **shebanq_passage*x*** databases, and read/write access for the **shebanq_web** database.

You need to create another passwords and store it safely: pwd-mysql-shebanq.
Maybe the database server is not the same as the application server. If they are the same, you can replace applicationserver by localhost

You will need the password pwd-mysql-root. Or alternatively, you need a non-root user with enough grants to perform the installation of the data, e.g.

```
mysql –u root –p
GRANT ALL PRIVILEGES ON `shebanq%`.* TO shebanq_admin@applicationserver';

mysql –u root –p



CREATE USER 'shebanq'@'applicationserver' IDENTIFIED BY'pwd-mysql-shebanq';
```

or (if the shebanq user already exists)

```
SET PASSWORD FOR 'shebanq'@'applicationserver' = 'pwd–mysql–shebanq';



GRANT SELECT ON shebanq_etcbc%.* TO `shebanq`@`%`;
GRANT SELECT ON shebanq_passage%.* TO `shebanq`@`%`;
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER ON shebanq_web.* TO
`shebanq`@`%`;
```

```
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, ALTER ON shebanq_note.* TO
`shebanq`@`%`;
FLUSH PRIVILEGES;
exit
```

EMDROS

Unpack the emdros software in an arbitrary directory, e.g. where you are now:

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

Configure for installation in /opt/emdros, build and install

```
cd emdros-3.4.0
./configure --prefix=/opt/emdros --with-sqlite3=local --with-mysql=yes --with-
swig-language-java=no --with-swig-language-python=yes --with-sqlite=no --with-
postgresql=no --with-wx=no --with-swig-language-csharp=no --with-swig-language-
perl=no --with-swig-language-ruby=no --with-bpt=no --disable-debug

make
make install
```

MYSQL DATA

Static, readonly data

shebanq_passage*x* databases (x = 4, 4b, etc)

Have been dumped in files and then bzipped2.

```
bunzip2 shebanq_passage*x*.sql.bz2
mysql -u root -p <shebanq_passage*x*.sql
```

(use pwd-mysql-root)

Dynamic, read-write data

Only the database *shebanq_web*. It contains users, sessions, queries, etc.

If you are installing the real shebanq, you have to consult additional documentation to get this data.

If you are installing your own shebanq, you can start from scratch, and this step is done.

EMDROS DATA

Static, readonly data

shebanq_etcbc*x* databases.

These can be loaded as follows.

Do this for all versions that you encounter in the installation sources, such as x=4, x=4b, etc.

If updating, drop the database first:

```
mysql -u root -p
  drop database shebanq_etcbc*x*;
  exit
```

```
bunzip2 shebanq_etcbcx.mql.bz2
/opt/emdros/bin/mql -b m -u root -e UTF8 -p 'pwd-mysql-root' <shebanq_etcbcx.mql
```

EMDROS-SHEBANQ ADDITIONAL CONFIG

Create a file with the password of the mysql user shebanq and another file with the mysql hostas follows:

```
mkdir /opt/emdros/cfg
vim /opt/emdros/cfg/mql.cfg
```

with contents the password pwd-mysql-shebanq.

```
vim /opt/emdros/cfg/host.cfg
```

with contents the mysql host, e.g. `localhost`.

WEB2PY

Create and store a new password for the web2py admin app (max 16 chars). We refer to this password as pwd-web2py-admin.
Here are instructions for setting up a production site and for a test site.
The main difference is a proper certificate versus a self-signed certificate.

Production site

Take care to have a proper certificate ready.
We have put a modified version of the script in the installation sources in such a way that it does not create a self-signed certificate, but uses a proper certificate.

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

Test site

We will let the web2py installer create a self-signed certificate for web2py.
Follow the script in the installation sources.

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

Production and Test site

Look at the lines specifying the certificate:

```
  SSLCertificateFile /etc/apache2/ssl/shebanq.ancient-data.org.cer
  SSLCertificateKeyFile /etc/apache2/ssl/shebanq.ancient-data.org.key
  SSLCertificateChainFile /etc/apache2/ssl/CHAIN-TERENA_SSL_CA_2.pem
```

and place these files in the indicated directories.

Answers to Questions:
Dialog Postfix Configuration: `Internet Site`

Test site
For the self-signed certificate, answer a number of questions in a way that corresponds to your situation.

Production and Test site
The script asks for admin password in the end. Take pwd-web2py-admin for this.

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

```
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.

Test site
Replace the newly create default site by the prefabricated one from the dropbox:

```
cp ~/shebanq-install/sites-available-default /etc/apache2/sites-available/
default
service apache2 restart
```

Do not forget to replace shebanq.mydomain.org by the name and domain chosen by you.

```
vim /etc/apache2/sites-available/default
```

Make changes.

```
service apache2 restart
```

Production Site
Finally save the web2py config for the apache default site as backup, assuming that backupdir points to the location where you store backups.

```
cp /etc/apache2/sites-available/default backupdir/sites-available-default

pushd /home/www-data/web2py/applications
rm -r welcome
rm -r examples
```

UPDATING WEB2PY
In order to update web2py itself, do this:

```
=====
cd /home/www-data
```

```
cp -r web2py web2py-bak

if [[ -e web2py_src.zip ]]; then
    rm web2py_src.zip
fi

wget http://web2py.com/examples/static/web2py_src.zip

service apache2 stop
unzip web2py_src.zip
mv web2py/handlers/wsgihandler.py web2py/wsgihandler.py
chown -R www-data:www-data web2py

service apache2 start
=====
```

If all works (but maybe it does not work!)

```
rm -rf /home/you/web2py
rm -r welcome
rm -r examples
```

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.
We put the shebanq app next to web2py, and link to it symbolically from the web2py/applications directory.

```
pushd /home/www-data
git clone https://github.com/etcbc/shebanq
pushd web2py/applications
ln -s /home/www-data/shebanq shebanq
popd
popd
cp web2py-routes.py /home/www-data/web2py/routes.py
chown -R www-data:www-data /home/www-data/web2py
chown -R www-data:www-data /home/www-data/shebanq
service apache2 restart
```

(In case of updating:

either on the commandline:

In order to ease updating, create a script

```
vim ~/update.sh
====
#!/bin/bash
```

```
# This a the script that you can run on the production server of SHEBANQ to upda
te the code and the data

# run it as follows:
#
# ./update.sh                               # if only code or docs has changed
# ./update.sh -d
 # if there are changes in the passage databases
# ./update.sh -de
# if there are changes in the emdros databases
#
# -de includes the actions for -
d and that includes the actions for no arguments.

service apache2 stop
cd /home/www-data/shebanq
if [ "$1" == "-de" ]; then
    echo "dropping etcbc database version 4"
    mysql --defaults-extra-file=/root/mysqldumpopt -
e 'drop database if exists shebanq_etcbc4;'
    echo "dropping etcbc database version 4b"
    mysql --defaults-extra-file=/root/mysqldumpopt -
e 'drop database if exists shebanq_etcbc4b;'
    echo "unzipping etcbc database dump for version 4"
    bunzip2 -f -k /home/dirkr/shebanq-install/x_etcbc4.mql.bz2
    echo "unzipping etcbc database dump for version 4b"
    bunzip2 -f -k /home/dirkr/shebanq-install/x_etcbc4b.mql.bz2
    echo "importing etcbc database for version 4"
    mql -n -b m -u root -p `cat /root/mqlimportopt` -e UTF8 < /home/dirkr/
shebanq-install/x_etcbc4.mql
    echo "importing etcbc database for version 4b"
    mql -n -b m -u root -p `cat /root/mqlimportopt` -e UTF8 < /home/dirkr/
shebanq-install/x_etcbc4b.mql
fi
if [ "$1" == "-d" -o "$1" == "-de" ]; then
    echo "loading passage database for version 4"
    mysql --defaults-extra-file=/root/mysqldumpopt < /home/dirkr/shebanq-
install/shebanq_passage4.sql
    echo "loading passage database for version 4b"
    mysql --defaults-extra-file=/root/mysqldumpopt < /home/dirkr/shebanq-
install/shebanq_passage4b.sql
fi
git pull origin master
cd /home/www-data/web2py
python -
c "import gluon.compileapp; gluon.compileapp.compile_application('applications/
shebanq')"
cd /home/www-data/shebanq
```

```
cp -R /usr/local/lib/python2.7/dist-packages/guppy modules
chown -R www-data:www-data /home/www-data/web2py
chown -R www-data:www-data /home/www-data/shebanq
sleep 2
service apache2 start
====
```
and give it execute permission:

```
chmod u+x ~/update.sh
```

Then you can update, after having logged in as root or sudoer, by just saying

```
./update.sh
```

end updating)

Check it out:
http://shebanq.mydomain.org - the shebanq site
https://shebanq.mydomain.org/shebanq/appadmin - shebanq maintenance
https://shebanq.mydomain.org/admin - web2py admin app.

[or, if you are doing the real shebanq site on ancient-data.org:
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 password pwd-web2py-admin.

NOTES
Using the web2py admin app, it is possible to *compile* the application. It speeds up the application.
However, if you update views, you have to recompile, because the compilation step freezes the view inclusions. The recompile step is coded in the *update.sh* script above.
By the way, updating shebanq from git can also be done through the web2py admin app. The disadvantage is that you can't do a restart of the webserver in this way.

SECURITY
Once the site is up and running, you can test the security by
https://www.ssllabs.com/ssltest/analyze.html?d=shebanq.ancient-data.org
It should at least get an A.
Here is a tutorial: https://raymii.org/s/tutorials/Strong_SSL_Security_On_Apache2.html (Needs Apache 2.4)

BACKUP and MAINTENANCE

It is recommended to backup the stuff that changes over time.
Or you can backup the whole server.
The important thing to back up is the database *shebanq*. In this database the user activities are consolidated.
All other databases function as data sources only.
Also backup the relevant config files, i.e. the config files that you modify with respect to the installation files below.

Database backup
* shebanq: at least every day, preferably every hour
* other databases: not needed
Write a script `/root/backup.sh` with contents, assuming that backupdir points to the location where you store backups.

```
=====
#!/bin/sh

dest="backupdir"
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_web | gzip > $dest/
shebanq_web.sql.gz
chmod go-rwx $dest/shebanq_web.sql.gz
mysqldump --defaults-extra-file=/root/mysqldumpopt shebanq_note | gzip > $dest/
shebanq_note.sql.gz
chmod go-rwx $dest/shebanq_note.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

====
[mysql]
password = 'pwd-mysql-root'
user = root

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

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

Session cleaning every hour (all expired sessions, and sessions without expiration if they are older than 600000 seconds, i.e. roughly a week)

```
crontab -e

0  * * * * /root/backup.sh
10 * * * * python /home/www-data/web2py/web2py.py -Q -S shebanq -M -R /home/www-data/web2py/scripts/sessions2trash.py -A -o -f -x 600000
```

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

APPENDIX A: Config file for Apache default site

We customize the `/etc/apache2/ports.conf` a little bit:

```
NameVirtualHost *:80
Listen 80

<IfModule mod_ssl.c>
    # If you add NameVirtualHost *:443 here, you will also have to change
    # the VirtualHost statement in /etc/apache2/sites-available/default-ssl
    # to <VirtualHost *:443>
    # Server Name Indication for SSL named virtual hosts is currently not
    # supported by MSIE on Windows XP.
    NameVirtualHost *:443
    Listen 443
</IfModule>

<IfModule mod_gnutls.c>
    Listen 443
</IfModule>
```

The contents of the file `/etc/apache2/sites-available/default` is contained in one of the installatation files, either
setup-web2py-ubuntu.sh (test server) or
setup-web2py-ubuntu-prod.sh (production server)

There is also a file default-ssl there, but we leave that disabled. We only do `a2ensite` for default.

APPENDIX D: config file for web2py

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