# **Database backup & migration**

Backup of the database can be performed via pg\_dump. This only takes care of the database layer. The files, namly models and timecourses, have to be backuped separetly.

## database dumps via pg\_dump

<http://www.postgresql.org/docs/8.1/static/backup.html>

pg\_dump dbname > outfile

psql dbname < infile

pg\_dump multiscale-galactose >

psql dbname < infile

Daily database backup is run in a cron job:

<http://www.cyberciti.biz/faq/how-do-i-add-jobs-to-cron-under-linux-or-unix-oses/>

User can install their own cron jobs using the crontab command.

Database backup is run nightly via cron script

./backupDatabase.sh

#### **Incremental backup and folder syncronization with Rsync**

rsync <options> <source directory> <target directory>

<http://www.mikerubel.org/computers/rsync_snapshots/>

<http://www.admin-magazine.com/Articles/Using-rsync-for-Backups/%28offset%29/2>

To edit your crontab file, type the following command at the UNIX / Linux shell prompt:

crontab -e

0 2 \* \* \* bash /home/mkoenig/multiscale-galactose/backupDatabase.sh

restart the cron service

sudo /etc/init.d/cron restart

# **Deploying Django**

Easiest way via Apache with mod\_wsgi module installed

<https://docs.djangoproject.com/en/dev/howto/deployment/wsgi/modwsgi/>

Install Apache

sudo apt-get install apache2 apache2-dev

Get the apache version

apache2 -v

Server version: Apache/2.2.14 (Ubuntu)

Server built: Jul 12 2013 13:39:34

localhost | 127.0.0.1

should give http\_response now

Where are the apache files and what is shown to the outside by Apache?

/var/www

/etc/apache2/httpd.conf

The aim of mod\_wsgi is to implement a simple to use [Apache](http://httpd.apache.org) module which can host any [Python](http://www.python.org) application which supports the Python [WSGI](http://www.wsgi.org) interface. The module would be suitable for use in hosting high performance production web sites, as well as your average self managed personal sites running on web hosting services.

<http://code.google.com/p/modwsgi/wiki/QuickInstallationGuide>

Source code tar balls can be obtained from:

<http://code.google.com/p/modwsgi/downloads/list>

After having downloaded the tar ball for the version you want to use, unpack it with the command:

tar xvfz mod\_wsgi-X.Y.tar.gz

Replace 'X.Y' with the actual version number for that being used.

./configure

make

sudo make install

Libraries have been installed in:

/usr/lib/apache2/modules

If you ever happen to want to link against installed libraries

in a given directory, LIBDIR, you must either use libtool, and

specify the full pathname of the library, or use the `-LLIBDIR'

flag during linking and do at least one of the following:

- add LIBDIR to the `LD\_LIBRARY\_PATH' environment variable

during execution

- add LIBDIR to the `LD\_RUN\_PATH' environment variable

during linking

- use the `-Wl,-rpath -Wl,LIBDIR' linker flag

- have your system administrator add LIBDIR to `/etc/ld.so.conf'

## Loading Module Into Apache

Once the Apache module has been installed into your Apache installation's module directory, it is still necessary to configure Apache to actually load the module.

sudo gedit /etc/apache2/mods-available/wsgi.load

and add line

LoadModule wsgi\_module /usr/lib/apache2/modules/mod\_wsgi.so

Activate the module and restart apache

sudo a2enmod wsgi

sudo service apache2 restart

\* Restarting web server apache2 AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1. Set the 'ServerName' directive globally to suppress this message [ OK ]

Configure an application

<https://code.google.com/p/modwsgi/wiki/QuickConfigurationGuide>

Create example application

def application(environ, start\_response):

status = '200 OK'

output = 'Hello World!'

response\_headers = [('Content-type', 'text/plain'),

('Content-Length', str(len(output)))]

start\_response(status, response\_headers)

return [output]

in

/usr/local/www/wsgi-scripts/myapp.wsgi

WSGIScriptAlias / /var/www/wsgi-scripts/hello.wsgi

<Directory /var/www/wsgi-scripts>

Require all granted

</Directory>

Set permissions

<http://www.slideshare.net/GrahamDumpleton/getting-started-with-modwsgi>

# **Server deploy information**

Deployment on the HITS development server

Necessary to compile all the dependencies for the django project with the necessary pyhton bindings

libsbml, libsedml, matplotlib, numpy, scipy

open issues are the database

- how to convert postgres 9.1/9.3 dump into 8.4?

ssh -x root@hitssv505

ip: 193.197.73.67

Apache VirtualHost running on 193.197.73.67:8002

libraries in:

/usr/lib/apache2/modules

config in:

/etc/apache2/httpd.conf

/etc/apache2/vhosts.d/mysite.conf

Here the VirtualHost information is setup on the server (here also all the dependent folders are stated)

restart server with:

apache2ctl restart

sudo service apache2 restart