Setup notes for SAC catalogue:

Tim Sutton 2009

Contents

1 Prepare your system

You need to be running Django >= 1.2.1 for the catalogue to work. Ubuntu Lucid and Debian Lenny ship with older versions so do a manual build. We walk through this setup using the python virtual environment system.

1.1 Create working dir

```
cd /opt
mkdir sac
cd sac
```

1.2 Setup python virtual environment

We install Python in a virtual environment on Ubuntu and Debian, to be able to install Django 1.2 separate from the "System Python" and avoid conflicts.

If you do not have the Python virtualenv software, get it with:

```
sudo apt-get install python-virtualenv
```

Now, start the Python virtual environment setup. We install Python in the "python" subfolder of the project directory and then activate the virtual environment.

```
virtualenv --no-site-packages python
source python/bin/activate
```

1.3 Install some development dependencies

We create a "dist" folder to save tarballs for installation.

Get the Python imaging tarball we need to install since it is not available using easy_install.

```
mkdir dist
cd dist/
wget http://effbot.org/downloads/Imaging-1.1.7.tar.gz
```

Untar and install the tarball just downloaded. We create a "src" folder in the projects directory for this purpose.

```
cd ..
mkdir src
cd src/
tar xzvf ../dist/Imaging-1.1.7.tar.gz
cd Imaging-1.1.7/
python setup.py install
cd ..
```

The next two products are installed with easy_install.

```
easy_install psycopg2
easy_install pip
```

1.4 Install django and required django apps

To install django, django authentication etc into our virtual environment do:

```
pip install -r sac_catalogue/requirements.txt
easy_install django-db-log
```

Then make sure the appropriate settings from djangodblog in settings.py.templ are deployed in your production settings.py

1.5 Further info on django registration

You may also want to read this:

http://devdoodles.wordpress.com/2009/02/16/user-authentication-with-django-registration/

if you want more info on how the registration stuff works.

Note: that you need to log in to the admin area of the site and change the domain name in the sites table from something other than 'example.com', otherwise the registration middleware will send the reminder with an incorrect url.

2 Source code Check out

Check out this folder using

```
svn co https://196.35.94.196/svn/trunk/sac_catalogue
cd sac_catalogue
```

Copy settings.py.template to settings.py and then modify settings.py as needed (probably you just need to set the eth adapter and db connection settings).

3 Database setup

Create the database using:

```
createlang plpgsql template1
psql template1 < /usr/share/postgresql-8.3-postgis/lwpostgis.sql
psql template1 < /usr/share/postgresql-8.3-postgis/spatial_ref_sys.sql
createdb sac
creatdb acs
```

3.1 For an empty database:

Sync the model to the db (dont do this is you plan to restore an existing db as explained in the next section):

```
python manage.py syncdb --database=default
```

And if you have the legacy acs catalogue do:

```
python manage.py syncdb --database=acs
```

The django fixtures included with this project should populate the initial database when you run the above command.

3.2 Restoring an existing database

Nightly backups are made on lion at:

```
/mnt/cataloguestorage1/backups/YEAR/MONTH/DAY/
```

To restore the backup do:

```
pg_restore sac_postgis_30August2010.dmp | psql sac
pg_restore acs_postgis_30August2010.dmp | psql acs
```

4 Setup apache

Make sure you have mod_expires and mod_deflate installed.

The assumption is that you are using name based virtual hosts and that the catalogue will run at the root of such a virtual host. Add to you apache site config:

```
<VirtualHost *>
  ServerAdmin tim@linfiniti.com
  ServerName it.sac.co.za
  ServerAlias www.it.sac.co.za
  LogLevel warn

  CustomLog /var/log/apache2/it.access.log combined
  # Possible values include: debug, info, notice, warn, error, crit,
  # alert, emerg.
  ErrorLog /var/log/apache2/it.error.log

  ServerSignature Off

  ScriptAlias /cgi-bin/ /usr/lib/cgi-bin/
  <Directory "/usr/lib/cgi-bin">
    AllowOverride None
    Options ExecCGI -MultiViews +SymLinksIfOwnerMatch
    Order allow,deny
    Allow from all
  </Directory>

  # This will match basically all addresses except
  # those with cgi-bin in them since we want
  # python proxy to work for cross site xhttprequests
  # see http://www.codinghorror.com/blog/archives/000425.html
  # for an explanation of the 'negative lookbehind' regex used here
  <LocationMatch "^/(?!cgi-bin).*$">
    SetHandler python-program
    PythonHandler django.core.handlers.modpython
    SetEnv DJANGO_SETTINGS_MODULE settings
    # see informix notes for why this is needed
    SetEnv INFORMIXSERVER catalog2
    PythonPath "['/opt/sac_catalogue/sac_test/'] + sys.path"
    PythonOption django.root /sac_test
    PythonDebug On
  </LocationMatch>

  # serve media files directly from apache for speed
  # see http://docs.djangoproject.com/en/dev/howto/deployment/modpython/#id1
  <Location "/admin_media">
      SetHandler None
  </Location>
  <Location "/media">
      SetHandler None
  </Location>
  Alias /admin_media /usr/lib/python2.5/site-packages/django/contrib/admin/media
  Alias /media /opt/sac_catalogue/sac_live/media
</VirtualHost>
```

5 Copy over the ribbon

There is a ribbon image that displays in the top left corner of the site that is used to convey version numbers etc. Since this may vary from deployment to deployment, you should copy over an appropriate ribbon e.g.:

```
cp media/images/ribbon_template.png media/images/ribbon.png
```

6 Install GEOIP data

GeoIP is used to resolve IP addresses to Lon/Lat. This directory needs the GeoIP lite dataset in it:

```
cd geoip_data
wget http://www.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz
gunzip GeoLiteCity.dat.gz‘
```

7 Check settings.py!

Go through settings.py (after first copying it from settings.py.templ if needed) and check all the details are consistent in that file.

8 Install proxy.cgi - note this will be deprecated

Some parts of this site use cross site XHttpRequests. This is not allowed in the spec (to prevent cross site scripting attacks) so to get around this you need to install a proxy cgi on the django hosting server *if the mapserver instance is on a different physical server*.

```
cd /usr/lib/cgi-bin
sudo wget -O proxy.cgi \
http://trac.openlayers.org/browser/trunk/openlayers/examples/proxy.cgi?format=raw
sudo chmod +x /usr/lib/cgi-bin/proxy.cgi
```

Once you have installed the proxy.cgi you need to configure it to tell it the list of allowed servers it can proxy for. This is to prevent it becoming an open relay on the internet. Edit /usr/lib/cgi-bin/proxy/cgi and change line 18 to look like this:

```
allowedHosts = [ ’196.35.94.243’,’lion’, ]
```

I also changed line 32 to look like this:

```
 url = fs.getvalue(’url’, "http://196.35.94.243")
```

so that the default proxy url is our wms server.

See http://faq.openlayers.org/proxyhost/all/ for more info...

9 Creating branches

When the code get stabilised to a certain point you should create a branch to mark that stable code base and then deploy it on the live server. To create the branch do e.g.:

```
svn cp https://196.35.94.196/svn/sac/python/sac_catalogue \
       https://196.35.94.196/svn/branches/catalogue_v1_beta3
```

Where: **v1** = version 1 **beta3** = the current status of that major version

10 Backup of the web server

sudo dd if=/dev/sdb | ssh definiens4 ”dd of=/cxfs/dd_backups/orasac1/orasac1_sdb_‘date +%a%d%b%Y‘.dd” sudo dd if=/dev/sda | ssh definiens4 ”dd of=/cxfs/dd_backups/orasac1/orasac1_sda_‘date +%a%d%b%Y‘.dd”

11 Creation of the ReadOnly db user

This should be done on the database server i.e. elephant

This user is required for mapserver access to some of the tables.

```
sudo su - postgres
createuser -S -D -R -l -P -E -e readonly
exit
psql sac
grant select on vw_usercart to readonly;
grant select on visit to readonly;
grant select on sensor to readonly;
\q
```

12 Optimal database configuration

To support the large number of recs tweak /etc/postgresql/8.3/main/postgresql.conf

```
# Changed by Tim as the sac db required more
max_fsm_pages = 500000
```

Then restart the db

```
sudo /etc/init.d/postgresql restart
```

13 set some file permissions

Apache user needs write access in avatars:

```
sudo chgrp www-data media/avatars
sudo chmod g+w media/avatars
```