

openerp.buildout

This is a buildout-based project to generate OpenERP instances for tests and development.

It is far from mature enough to be used in production, but feel free to send patches to help me improve this situation.

This was tested with Debian Squeeze and is likely to work on Ubuntu.

All the commands below should be performed with your normal user. Each time we need extra privileges we prefix our commands with `sudo`.

I aim to make this script less messy and sysadmin-friendly but this goal often conflicts with making a fully-automated script. However I think these two goals can be reached by making `openerp.buildout` a little bit more modular. But I'm not sure how, so I need ideas and contributions from people using it in real life.

Install OpenERP on a local machine

1. Install required dependencies with your favorite package installer.

- Here is for Debian:

```
$ sudo apt-get update
```

```
$ sudo apt-get install sudo python-dev gcc bzip2 subversion git postgresql libxslt1-c
```

2. Get a copy of this buildout template:

```
$ git clone git://github.com/kdeldycke/openerp.buildout.git
```

```
$ cd ./openerp.buildout
```

3. Initialize the buildout environment:

```
$ python ./bootstrap.py --distribute
```

4. By default OpenERP 6.0 will be installed for development. If you need another version, or the production profile, edit the `buildout.cfg` file and update the `extend` parameter of the `[buildout]` section accordingly.

5. Run buildout itself:

```
$ sudo ./bin/buildout
```

6. Start the PostgreSQL server:

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

7. Now you can launch the OpenERP server and web client:

```
$ sudo /etc/init.d/openerp-server restart
$ sudo /etc/init.d/openerp-web-client restart
```

8. To play with your OpenERP instance, all you have to do is to point your browser to:

```
http://127.0.0.1:8080
```

9. And to monitor everything's all right, keep an eye on the logs:

```
$ tail -F /var/log/openerp/openerp-*.log
```

10. Finally to let the server and the web client starts with the machine, do:

```
$ sudo update-rc.d openerp-server defaults 90
$ sudo update-rc.d openerp-web-client defaults 91
```

Command helpers

Or how to make your daily usage of buildout painless.

bin/buildout

This is the command to run to build a new environment.

This is also the command you call to update your environment.

In fact you can safely call it whenever you want. buildout implement an internal mechanism that keeps tracks of what it installed and configured before, that's why it's safe. And that's also why you can re-call it if it crashed in the middle of its execution.

bin/openerp-server.sh & bin/openerp-web-client.sh

There are script helper available in `./bin` to let you launch OpenERP's server (`./bin/openerp-server.sh`) and web client (`./bin/openerp-web-client.sh`) easily with the right user and configuration.

For example, if you need to update all modules of a database, you can use the following command:

```
$ ./bin/openerp-server.sh -d my_database_id -u all
```

bin/openerp-shell

All those scripts are based on `./bin/openerp-shell` which is a simple wrapper around your system's Python interpreter, only it add under its scope all modules locally installed by this buildout. So if you're looking to use whatever Python module buildout install for you, forget about your system's Python interpreter and use `./bin/openerp-shell` instead, see:

```
$ python
Python 2.6.6 (r266:84292, Dec 26 2010, 22:31:48)
[GCC 4.4.5] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> import mako
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
ImportError: No module named mako
>>>

$ ./bin/openerp-shell
>>> import mako
>>> repr(mako)
"<module 'mako' from '/home/openerp.buildout/eggs/Mako-0.4.2-py2.6.egg/mako/__init__.py">
>>>
```

Configuration extensions

OpenERP GTK client

To install the GTK fat client, install first some dependencies:

```
$ sudo apt-get install xorg ghostscript graphviz python-hippocanvas python-libxml2 python
```

Then add `profiles/openerp-gtk-6.0.cfg` to the `extend` parameter of the `[buildout]` section of `buildout.cfg`.

When the buildout finish the installation, before running the GTK client itself, you have to launch X.org first:

```
$ sudo startx &
$ sudo su openerp -c "./bin/openerp-gtk-client-shell ./parts/openerp/gtk-client/bin/openerp-gtk-client"
```

Aeroo

Aeroo is a reporting engine for OpenERP.

To let this buildout install and configure it, you first have to install OpenOffice (or LibreOffice) and some other dependencies:

```
$ sudo apt-get install openoffice.org python-openoffice python-cairo-dev
```

Then add `profiles/aeroo.cfg` to the `extend` parameter of the `[buildout]` section of `buildout.cfg`, as demonstrated in the `custom.cfg` file.

Finally, you have to start the office server in a headless mode:

```
$ sudo /etc/init.d/office-server restart
```

And to let the office starts with the machine, do:

```
$ sudo update-rc.d office-server defaults 90
```

Apache

If you want to serve OpenERP web client through Apache, a configuration file can be automatically generated.

First install Apache itself:

```
$ sudo apt-get install apache2
```

Then add `profiles/apache.cfg` to the `extend` parameter of the `[buildout]` section of `buildout.cfg`, and set the `web-domain` variable to your public domain name, as demonstrated in the `custom.cfg` file.

Troubleshooting

Locale

You may encounter this issue when Bazaar commands are run:

```
bzr: warning: unsupported locale setting
bzr could not set the application locale.
Although this should be no problem for bzr itself,
it might cause problems with some plugins.
To investigate the issue, look at the output
of the locale(1p) tool available on POSIX systems.
```

This can be fixed by forcing your locale to more sane defaults.

Here is how I did it on Debian Squeeze (source):

```
$ echo "LANG=en_US.UTF-8" >> /etc/default/locale
$ echo "en_US.UTF-8 UTF-8" >> /etc/locale.gen
$ locale-gen
```

Then I had to log out and back in to have the environment variables set to the new defaults.

And here is another way of fixing this local issue (thanks Xavier for the tip !):

```
$ apt-get install --reinstall language-pack-fr
$ dpkg-reconfigure locales
```

Proxy

Behind a proxy ? Just set the appropriate environment variables to let Subversion and Bazaar pass through it:

```
$ export http_proxy=http://10.0.2.1:8080
$ export https_proxy=http://10.0.2.1:8080
```

Note that some versions of Bazaar are affected by a bug which doesn't let you pass a proxy. In my case, the `bzr-2.1.2-1` package available on my Debian Squeeze was affected by such a bug. To fix this, I just applied the patch proposed at <http://code.edge.launchpad.net/~lifeless/bzr/bug-558343-wrong-host-with-proxy/+merge/24938> .

Install OpenERP in the cloud (work in progress)

1. Get a copy of this buildout template:

```
$ sudo apt-get install git
$ git clone git://github.com/kdeldycke/openerp.buildout.git
$ cd ./openerp.buildout
```

2. Initialize the buildout environment:

```
$ python ./bootstrap.py --distribute
```

3. Edit the `buildout.cfg` file and uncomment the line referring to `profiles/cloud.cfg`. Update the `[amazon-instance]` section with your Amazon AWS key and secret.

4. Run buildout itself:

```
$ ./bin/buildout -N
```

5. Then we have to create an instance in the cloud:

```
$ ./bin/hostout amazon-instance create
```

6. Then we have to initialize the remote server:

```
$ ./bin/hostout amazon-instance bootstrap
```

7. Deploy to EC2:

```
$ ./bin/hostout amazon-instance deploy
```

8. To test, we can run our remote OpenERP shell:

```
$ bin/hostout amazon-instance run bin/openerp-shell
```

9. And finally, once you don't need your cloud instance, you can destroy it:

```
$ ./bin/hostout amazon-instance destroy
```

To fine-tune your cloud and choose the right instance, you can run these commands:

```
$ ./bin/hostout amazon-instance list_providers
$ ./bin/hostout amazon-instance list_nodes
$ ./bin/hostout amazon-instance list_sizes
$ ./bin/hostout amazon-instance list_images
```

TODO

- Group all system initialization commands in one separate buildout config file. Basically everything that has to be run as `root` will be moved there. The new `buildout.cfg` will then be able to be run without `root` privileges.
- Auto-update all modules of all databases on update.
- Don't call the patch command directly, use <http://pypi.python.org/pypi/collective.recipe.patch>
- Consider using and contributing to <http://pypi.python.org/pypi/anybox.recipe.openerp>
- Get inspiration from <http://github.com/kalymero/OpenERP-Buildout> ?

- Automate locale fix ?
- Use mr.developer to download OpenERP components from bazaar ? It looks like mr.developer is smarter and can speed up the download process...
- Or look at <http://pypi.python.org/pypi/checkoutmanager>
- Explore the use of OpenERP 6.1 web client as a module (see: <http://twitter.com/#!/fpopenerp/status/111111111111111111>)
- Worth looking at isotoma.recipe.facts for extra info and automation: <http://pypi.python.org/pypi/isotoma>
- Add a way to generate a big archive of all components used in that OpenERP build. This may be useful for customer's project delivery.
- It maybe worth checking this: <http://pypi.python.org/pypi/cloudinitd>

Author

- Kevin Deldycke - kevin@deldycke.com

Contributors

- Xavier Fernandez

License

This code is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, version 2.

This program is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

For full details, please see the file named COPYING in the top directory of the source tree. You should have received a copy of the GNU General Public License along with this program. If not, see <http://www.gnu.org/licenses/>.

Embedded external projects

This tool uses external softwares, scripts, libraries and artworks:

Buildout's bootstrap.py
Copyright (c) 2006, Zope Corporation and Contributors
Distributed under the Zope Public License, version 2.1 (ZPL).
Source: <http://svn.zope.org/repos/main/zc.buildout/trunk/bootstrap/bootstrap.py>