

Cubicweb installation

07-Jun-2012

This is a quick guide to get you jump started to install cubicweb development environment in a clean linux server, in my case I am using a Debian Squeeze 32 Bits.

Quick note: anytime you see the “#” at the beginning of the command line it means that command should be executed as a root user.

First thing to do is to create the user we are going to be using during a “normal” operation of the operating system, in this case I have already created a user called “carlos”, if you don’t have one already please go to step 1 in this document, otherwise go to step 2.

1.- Create a personal user.

```
# adduser “myuseraccount”
```

You will be asked for Password, Full Name, Room Number, Phones, etc, and if the information you have filled is correct.

Making this user a sudo user will ease our installation process, so we don’t have to use the “root” user to download and install dependencies packages and change back and forward between a normal and root user. If the users is not already a sudo user go to step 2 in this document.

2.- Make this user a sudo user, so we can install packages in the server.

2.1.- In case you have not installed sudo package : # apt-get install sudo

2.2.- # visudo

2.3.- Look for: root ALL=(ALL) ALL and do the same for the user you want to add to sudo users in my case I am using “carlos” account. e.i. carlos ALL=(ALL) ALL

3.- Now exit from the root account if you still there and change to the user you are going to be working with, in my case, “carlos” account.

In case you followed our previous tutorial to install Debian you should have PostgreSQL installed already, but if you did not, then please go to step 4 and install it.

4.- Install Postgresql

```
# sudo apt-get install postgresql
```

We'll be using mercurial (<http://mercurial.selenic.com/>) as our source control, if you don't have it installed then go to step 5 please.

5.- Install Mercurial

5.1.- Install dependencies: `sudo apt-get install build-essential gcc python-dev` (We are going to be asked to insert the Debian CD1, so please have it at hand).

5.2.- Install gettext dependency: `sudo apt-get install gettext`

5.3.- Download mercurial package and save it into `/home/"myuseraccount"/src/mercurial` folder. e.g. `/home/carlos/src/mercurial`

5.4.- Copy package to `/usr/local/src` folder. `sudo cp mercurial-package /usr/local/src`

5.5.- Change to the path `/usr/local/src` and uncompress the package. `sudo tar -xvf mercurial-2.1.tar.gz`

5.6.- Install docutils: `sudo apt-get install python-docutils`

5.7.- Change to the folder `mercurial-2.1` and read README file and then run:

5.7.1.- `sudo make all`

5.7.2.- `su -c "make install"` (you will be asked for the root password).

5.7.3.- `hg version` (this should return the version information about the installation you have just done).

5.8.- In your "**home**" directory e.g. `/home/carlos/`, create the hidden file `.hgrc`, which hold all the main mercurial configuration. Add this lines:

```
[ui]
```

```
username=Your Name and Your Email Account. e.g. Carlos Balderas
```

```
<carlos.balderas@gmail.com>
```

```
merge=merge
```

In our case we stick with hgforest extension to handle multiple repositories at once, you can choose what is best for you, so step 6 is optional.

6.- Install Mercurial Forest

6.1.- Go to your "source" folder `/home/"myuser"/src/`. e.g. `/home/carlos/src/`

6.2.- Use mercurial to clone the forest code.

```
hg clone https://bitbucket.org/andyhelp/hgforest-crew hgforest
```

6.3.- Add this lines to your `~/.hgrc` file.

```
[extensions]
```

```
hgext.forest=/home/"myuser"/src/hgforest/hgext/forest.py
```

```
# or, if forest.py is not in the hgext dir:
```

```
# forest=/path/to/forest.py
```

6.4.- Check that you have typed the correct path by doing: `hg` and enter, if you see the

commands like `fcclone`, `fpush`, `fstatus`, then you are on the right way.

You will be needing an user with administrator rights in your postgresql server to create your cubicweb instances, in case you don't have already one please go to step 7.

7.- PostgreSQL configuration.

7.1.- Change to user "postgres" (default user created at the postgresql installation time).

```
sudo su postgres
```

7.2.- Create the user that will have privileges in postgresql databases.

```
createuser -s -P cwusername (e.g. createuser -s -P carlos)
```

7.3.- Change to the normal user, in my case "carlos". By doing "exit" you can achieve this.

8.- Cubicweb dependencies

Some important notes:

1.- My recommendation is to install from source most of the dependencies packages used by cubicweb, but in most of the packages I wrote the package you can install using `apt-get` command.

2.- In order to keep things standardize my recommendation is to use your **src** folder in your home directory (e.g. `/home/carlos/src/`) to keep all the downloaded sources, and the use this other path **/usr/local/src** as your installation path. Again, this is up to you to do so :).

3.- At the end of most installation instructions listed below, I give you 2 options: For Package and For `apt-get`, you must choose only one of them.

4.- If you choose to install from source, you should first download the source from the official web site, looking the package type "source", regularly with extension `tar.gz` or `tar.bz2` in some cases, then you can place it anywhere you want in my case I use (`/home/carlos/src/"mypackagefolder"`), after that, you are ready to install it, What I do is copy the package to the path **/usr/local/src/** (anything done inside here needs root rights, so a good option is to use `sudo`) then, decompress them there, go to the created folder and follow the "For Package" instructions.

5.- In a very few packages listed here, you will find only one way to install the package, this is just my recommendation because I found more difficult to do it in other way, if you are a more experienced linux user you can give a try anyway you prefer :).

6.- If you happen to choose to install using `apt-get` install, everything would be more smooth, but in some cases you'll not get the newest version of certain packages, but it should work well too.

8.1.- sudo apt-get install gettext

The GNU gettext package provides a set of tools and documentation for producing multi-lingual messages in programs. Tools include a set of conventions about how programs should be written to support message catalogs, a directory and file naming organization for the message catalogs, a runtime library which supports the retrieval of translated messages, and stand-alone programs for handling the translatable and the already translated strings. Gettext provides an easy to use library and tools for creating, using, and modifying natural language catalogs and is a powerful and simple method for internationalizing programs.

8.2.- yapps2-2.1.1-17.1 <http://pypi.python.org/pypi/Yapps2/>

Yapps (Yet Another Python Parser System) is an easy to use parser generator that is written in Python and generates Python code. Yapps is simple, is easy to use, and produces human-readable parsers. It is not fast, powerful, or particularly flexible. Yapps is designed to be used when regular expressions are not enough and other parser systems are too much -- situations where you may write your own recursive descent parser.

*** For Package: sudo python setup.py install**

*** For apt-get: sudo apt-get install yapps2**

8.3.- graphviz / <http://www.graphviz.org>

What is Graphviz?

Graphviz is open source graph visualization software. Graph visualization is a way of representing structural information as diagrams of abstract graphs and networks. It has important applications in networking, bioinformatics, software engineering, database and web design, machine learning, and in visual interfaces for other technical domains.

*** For apt-get: sudo apt-get install graphviz**

8.4.- Graphviz dependencies

*** For apt-get: sudo apt-get install libexpat1-dev libgraphviz-dev**

8.5.- python-pygraphviz <http://pypi.python.org/pypi/pygraphviz>

PyGraphviz is a Python interface to the Graphviz graph layout and visualization package. With PyGraphviz you can create, edit, read, write, and draw graphs using Python to access the Graphviz graph data structure and layout algorithms. PyGraphviz is independent from NetworkX but provides a similar programming interface.

*** For Package: sudo python setup.py install**

*** For apt-get: sudo apt-get install python-pygraphviz /**

If you think your installation is correct you will need to manually change the include_path and library_path variables in setup.py to point to the correct locations of your graphviz installation.

The current setting of library_path and include_path is:

library_path=None

include_path=None

8.6.- python-simplejson <http://pypi.python.org/pypi/simplejson/>

Simple, fast, extensible JSON encoder/decoder for Python

- * **For Package:** `sudo python setup.py install`
- * **For apt-get:** `sudo apt-get install python-simplejson`

8.7.- python-docutils <http://docutils.sourceforge.net/>

You probably have this one because mercurial needs it.

Docutils is a modular system for processing documentation into useful formats, such as HTML, XML, and LaTeX. For input Docutils supports reStructuredText, an easy-to-read, what-you-see-is-what-you-get plaintext markup syntax.

- * **For Package:** `sudo python setup.py install`
- * **For apt-get:** `sudo apt-get install python-docutils`

8.8.-python-lxml

xml is a Pythonic, mature binding for the libxml2 and libxslt libraries. It provides safe and convenient access to these libraries using the ElementTree API.

It extends the ElementTree API significantly to offer support for XPath, RelaxNG, XML Schema, XSLT, C14N and much more.

***The easiest way to install this is using apt-get install, it is a little bit complicated to compile this package*

- * **For apt-get:** `sudo apt-get install python-lxml`

8.9.- python-twisted-web2 <http://twistedmatrix.com/trac/>

Twisted is an event-driven networking engine written in Python and licensed under the open source [MIT license](#)

- * **For Package:** `sudo python setup.py install`
- * **For apt-get:** `sudo apt-get install python-twisted-web2`

8.10.- pyro <http://pypi.python.org/pypi/Pyro4/>

Pyro stands for PYthon Remote Objects. It is an advanced and powerful Distributed Object Technology system written entirely in Python, that is designed to be very easy to use.

- * **For Package:** `sudo python setup.py install`
- * **For apt-get:** `sudo apt-get install pyro`

8.11.-Libraries needed.

For apt-get: `sudo apt-get install libpq-dev`

8.12.- python-psycopg2 <http://pypi.python.org/pypi/psycopg2>

psycopg2 is a PostgreSQL database adapter for the Python programming language. psycopg2 was written with the aim of being very small and fast, and stable as a rock.

psycopg2 is different from the other database adapter because it was designed for heavily multi-threaded applications that create and destroy lots of cursors and make a conspicuous number of concurrent INSERTs or UPDATES. psycopg2 also provide full asynchronous operations and support for coroutine libraries.

*** For Package: sudo python setup.py build, sudo python setup.py install**

*** For apt-get: sudo apt-get install python-psycopg2**

8.13.- python-setuptools <http://pypi.python.org/pypi/setuptools#downloads>

Download, build, install, upgrade and unistall Python packages -- easily!.

*** For Package: sudo python [setup.py](#) install**

*** For apt-get: sudo apt-get install python-setuptools**

8.14.- python-dateutil <http://pypi.python.org/pypi/python-dateutil>

The dateutil module provides powerful extensions to the standard datetime module, available in Python 2.3+.

*** For Package: sudo python [setup.py](#) install**

*** For apt-get: sudo apt-get install python-dateutil**

8.15.-python-vobject <http://pypi.python.org/pypi/vobject#downloads>

Parses iCalendar and vCard files into Python data structures, decoding the relevant encodings. Also serializes vobject data structures to iCalendar, vCard, or (experimentally) hCalendar unicode strings.

*** For Package: sudo python [setup.py](#) install**

*** For apt-get: sudo apt-get install python-vobject**

8.16.- python-psyco <http://psyco.sourceforge.net/psycoguide/sources.html>

**** Install subversion - version control system to download, please follow the instructions listed in the official web site.**

Psyco shows that it is possible to execute Python code at speeds approaching that of fully compiled languages, by "specialization". This extension module for the unmodified interpreter accelerates user programs with little or not change in their sources, by a factor that can be very interesting (2-10 times is common).

*** For Package: sudo python [setup.py](#) install**

8.17.- For apt-get: sudo apt-get install postgresql-contrib-8.4 (check the last numbers in the name of the package so you can use the correct ones according to your postgresql version)

8.18.- For apt-get: sudo apt-get install postgresql-plpython-8.4 (check the last numbers in the name of the package so you can use the correct ones according to your postgresql version)

8.19.- passlib <http://pypi.python.org/pypi/passlib>

Passlib is a password hashing library for Python 2 & 3, which provides cross-platform implementations of over 20 password hashing algorithms, as well as a framework for managing existing password hashes. It's designed to be useful for a wide range of tasks; from verifying a hash found in /etc/shadow, to providing full-strength password hashing for multi-user applications.

*** For Package: sudo python [setup.py](#) install**

9.- Clone Cubicweb repository.

Go to your source "src" directory e.g. /home/carlos/src

9.1.- Create the folder fcubicweb and move to it.

9.2.- hg clone <http://hg.logilab.org/cubicweb>

9.3.- python cubicweb/bin/clone_deps.py (This will clone some cubes made by logilab company so you can start testing them right away).

10.- Set Cubicweb configuration.

10.1.- Edit your .bashrc hidden file located in your home directory. e.g. /home/carlos/.bashrc and place the next lines:

```
export PYTHONPATH=/home/"pathto..."/fcubicweb/  
export CW_CUBES_PATH=/home/"pathto..."/fcubicweb/cubes  
alias cubicweb-ctl='~/src/fcubicweb/cubicweb/bin/cubicweb-ctl'
```

10.2.- Execute the command "source .bashrc" to reload the changes.

11.- Clone hgview tool.

Go to your source "src" directory e.g. /home/carlos/src

11.1.- hg clone <http://hg.logilab.org/hgview>

11.2.- Edit your .bashrc hidden file located in your home directory. e.g. /home/carlos/.bashrc and add the next line:

```
alias hgview='~/src/hgview/bin/hgview'
```

11.3.- Execute the command "source .bashrc" to reload the changes.

11.4.- sudo apt-get install python-qscintilla2

11.5.- sudo apt-get install pyqt4-dev-tools

All set, now you can go to your command line and type cubicweb-ctl list, to see what happens.