



from
June 2015
Strategy Cookbook



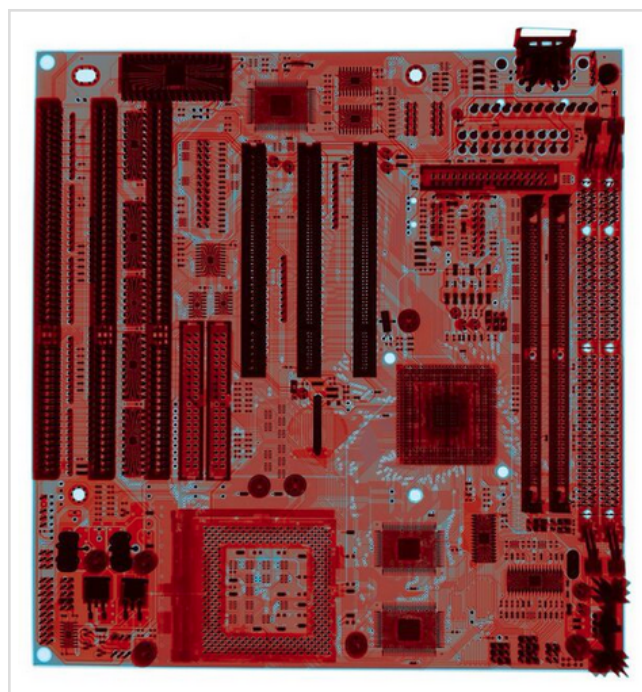
DRUPAL IN DEBIAN

System Integration for Drupal

Gunnar Wolf

0

As far as how to install and maintain Drupal, there is no need to dig further. Installing Drupal in any hosting provider is simply a matter of decompressing a tarball; Drush on the command line gives the seasoned sysadmin a wealth of administrative aids. However, when considering integration with the system as a whole, there is ample room for improvement. Drupal cannot exist outside of a given environment: the PHP version used, the way it is integrated into its host operating system, the modules for connecting to the database, and the database engine itself. Ideally, they will all make up a coherent entity, with a single, unified administrative logic. Looking at each component individually would quickly lead to madness, especially for the sysadmin, who has to keep the whole stack secure and updated.



Debian

Debian is one of the earliest freeware distributions built around Linux. By the time this article is printed, Debian 8 (codenamed *Jessie*) should have been released, with over

35,000 *binary packages* (that is, independent programs), and lives according to the distribution's motto, *the universal operating system*. It runs from the smallest embedded devices to the largest workstations, across many different hardware architectures.

System-wide Integration

One of Debian's strongest merits – what has made it a lively, energetic community with a sound technological platform and projection into the future – is its policy. Despite the amount of available packages, they are all standardized: They are all configured in the same location, follow the same layout logic, and are guaranteed not to clash with one another. But where this policy shines most brightly is when it is applied to keeping things optimally administered. Debian provides security support throughout the stable release cycle, not just easing the system's installation. As was already stated, our Drupal installs involve quite a bit beyond *just* Drupal itself. So, on a freshly installed system, the task of installing

Drupal is just a matter of running:

```
# aptget install drupal7
```

And all of the necessary software (that is, an HTTP server, the PHP programming language and execution environment, a relational database engine, and the needed glue between them) will be installed. To apply all of the pending security and reliability fixes across the entire system, run the command:

```
# aptget update && aptget upgrade
```

That's all that's needed to get every component in the system up to date.

The Debian Drupal installation is multisiteaware. This means that all of the sites Drupal will respond to should be configured from the same location. If your host will serve both *example.com* and *anotherexample.org*, you only need to create the `/etc/drupal/7/sites/example.com` and `/etc/drupal/7/sites/anotherexample.org` directories and put the sites' configuration files there. All of the codebase will be shared, installed only once in the `/usr/share/drupal7` directory.

This has an interesting advantage securitywise when compared with what I have seen at most shared hosting providers. As all the Drupal code belongs to root, any attacker that manages to subvert Drupal's security – or any of the installed modules' security – will not have enough privileges to modify your site's code, and will thus have a harder time leaving a backdoor or modifying your site's behavior for their interests. Even if they got a local privilege escalation exploit, finding their misdeed will be easier: Debian ships cryptographic signatures for all of its files. By simply running the following command, any file that was modified will be reported:

```
$ debsums drupa17
```

Handling Modules

The Drupal ecosystem is very rich in thirdparty code: almost 30,000 modules and over 2,000 themes. Packaging them all for Debian is plainly unfeasible, but we have a tool – `dhmakedrupa1` – that not only packages modules and themes pointed to it, but processes dependency handling as well. This way, even having a complex, multisite and multiserver deployment, it's easy to deliver code with all of the characteristics mentioned in the previous section.

Version Stability

Just as, within Drupal, all of the related PHP and Javascript libraries are frozen prior to a release and not hanged at that point (to avoid breaking internal stability throughout the life cycle of a stable release), packages in Debian are frozen and do not get updated when new versions come out. But in order to keep security tracking at the level required by our users, all important fixes get backported to the corresponding release. For example, Debian 7 (frozen gradually since June 2012 until its release in May 2013) shipped with Drupal 7.14. But that does not mean that Debian's Drupal package went on for its three years of life unpatched: While feature releases were not incorporated, as you can see in our public Git repository, all bugfix releases were.

What About Drupal 8?

Now... With all the hype set in the future, it might seem striking that throughout this article I only talked about Drupal 7. This is because Debian seeks to ship only production ready software: as of this writing, Drupal 8 is still in beta. Not too long ago, we still saw internal reorganizations of its directory structure.

Once Drupal 8 is released, of course, we will take care to package it. And although Drupal 8 will not be a part of Debian 8, it will be offered through the *Backports* archive, following all of Debian's security and stability requirements.

Wrapping Up

Of course, I understand the workflow and tools mentioned here are not for every person and situation. Drupal developing is clearly served better by the “traditional” way of installing Drupal. However, for administering a large scale Drupal installation, operating system integration through Debian might be just what you need!

Image: "X-ray of computer motherboard (<https://flic.kr/p/6VFQU4>)" by tpmartins (<https://www.flickr.com/photos/tjmartins/>) is licensed under CC BY 2.0 (<https://creativecommons.org/licenses/by/2.0/>)

Share your thoughts