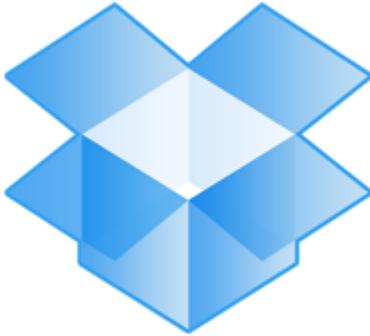


Using Dropbox with Unix



Although last week saw some pretty exciting developments in the cloud storage (Google Drive announcement and SkyDrive free 25Gb space), the truth is that **Dropbox** is still the king of the cloud storage hill – it's hands down the easiest to use and integrate.

I've been a **Dropbox** user for a few years now, but have started using it actively only in the last 12 months or so. It's been an invaluable tool for me thanks to its integration with 1Password, the password tool of my choice. **Dropbox** also helps with lots of day-to-day tasks and that's why I decided it's time to share some of the tips.

Having used **Dropbox** extensively on Windows systems (XP on laptop and Win7 on desktops), I've recently moved on to using **Dropbox** with my Mac OSX desktop and Linux hosting.

So here are the top tips for using **Dropbox** with Unix – each one does wonders for me and so I hope you like them as well.

Important: If you're not a **Dropbox** user yet, [please use this link to sign up](#) – it means I'll get a small bonus (extra 500MB to my free account) for referring you.

Storing all the common apps and tools in Dropbox

Dropbox is really smart when it comes to uploading your files

into the cloud storage and making it universally accessible across all the devices that you choose to pair with your **Dropbox** account.

One thing I particularly like using **Dropbox** for is storing the latest (or sometimes not the latest but verified to be fully working) versions of apps and tools I find handy to have on my desktops. In addition to having installers for all your favourite tools available on each workstation, **Dropbox** account is also handy for simply storing all the necessary software in one location. When traveling, for example, I can open my **Dropbox** account and safely download the exact version of a particular tool that I need. It saved me a lot of time because I don't have to go to each website and search for that download link.

Synchronizing scripts and config files between hosting systems

I have a dedicated server and use it for running a number of Ubuntu VMs. I've created a separate **Dropbox** account for my hosting needs, and this means that I now have 2.5Gb of space available for my VMs to exchange files or store immediate backups. Because **Dropbox** takes care of synchronizing all the content (and it has a LAN sync feature meaning VMs transfer files directly to each other instead of uploading back to the Dropbox site), it's super easy and super fast to have a particular script updated and deployed to multiple systems.

I'm not quite there yet with actually running stuff like important automation or whole websites straight from **Dropbox** directory, but I use it for deploying scripts and configs all the time – once I get something working properly on one VM, I can then hope from one system to another and run the same set of commands against the files which are synchronized by **Dropbox**.

Transferring files to and from my hosting

This is a very recent addition to the things I do with **Dropbox**, but it's an incredibly useful one. Having setup a separate **Dropbox** account for hosting, I shared one of the folders with my personal **Dropbox** account, and this means that transferring any files to and from my hosting had gotten to be this much easier. By putting a file into a local directory on my desktop, I have it accessible accross all the VMs on my hosting within seconds.

Likewise, if I'm reading logs or working on updating a particular config file, I can always copy it into **Dropbox** directory and have it synced back to my desktop.

Prior to this setup I had to rely on scp (passwordless logins using passphrase), and although it was pretty convenient to use, **Dropbox** approach is much more robust. Because files appear to be local, you get to work with them and manage directories as you like. You don't have to remember the directory tree structure or follow any naming conventions – your files are the same across all the systems and you don't have to remeber to always sync.

Keeping backups of DBs or websites in Dropbox

Since majority of my websites are publicly available blogs, I don't consider most of the backups to be a sensitive information. To be clear, I don't store my passwords (wp-config.php file or htpasswd files) in **Dropbox** account, but everything else gets copied into it as a first level backup. I also have been doing automatic backups to **Amazon's S3** storage for about 5 years now, this means I can recover from most disasters quickly enough.

The reason **Dropbox** wins is because I don't have to pay for each minor transfer or for storing an extra gigabyte or two –

and yes, every little helps even though **Amazon's** services are quite affordable. Another major reason I started doing backups to **Dropbox** is because it's a local directory – I don't have to use any extra tools to access all the backups in a simple directories/files structure. With **Amazon's S3** it's also possible but setup is not as trivial.

Using Dropbox for controlling Unix systems remotely

With a few minutes and a really simple script, it's possible to setup your own mission control for all the VMs in your hosting.

For example, if you create a cronjob which looks for a particular file, you can control which DB server your systems will connect to or which directory you'll get the latest important log file copied into.

I'm also playing with services management based on the **Dropbox** account. If there's a file present, I keep a particular service running. As soon as the file is gone, my cronjob gracefully stops the service. A slightly more sophisticated approach involves storing services names and system names associations in a Dropbox synchronized file – this allows for more flexibility as I can specify which service I want to be running on which nodes.

Sure enough, this isn't the most straightforward way to manage your system, but such an approach can be used on the go from your iPhone. For example, I can restart a webserver by just touching a file from my iPhone, while previously I would have to find the nearest computer I trust, download SSH client, connect to the box and only then fix the problem.

Have I convinced you enough? Did you like any tips, or do you have some more perhaps? Let me know in the comments section!

pS: if you don't have a **Dropbox** account or perhaps if I persuaded you to create a separate one for your hosting – [please use this link](#) so that I get some extra Dropbox space for referring. Thanks!