

My webserver has died! How do I bring it back online in a hurry?

The obvious answer to this is to restore your backup onto a spare server. Or if it's just a drive failure, the answer would be to restore your backup onto your spare drive. But since you're reading this instead of restoring your backup, let's assume that the situation is not quite so simple.

So here's a hypothetical situation:

You've just launched a new website and your web server has decided that now is the time for a drive failure. The rest of the server hardware is just fine, but you don't have any spare drives lying around and you don't want to venture out into the LA traffic to buy a new drive. Fortunately, you do have a recent backup of your data. Or if not, your disk is not quite dead yet and you've read my other article on [how to make an emergency backup](#) .

(There's more than one way to go about this... but I'm just going to detail what I've done before)

Some simplifying assumptions:

- Your server was running Debian Linux... this is actually sort of flexible... the general idea still works with other Linux distros, but you may need to tweak some paths or configuration files
- You have (or can borrow) another server that has spare disk space and is capable of doing NFS exports (this server is currently in use and you can't just switch it to the IP address of the server with the failed drive)

Here's the plan:

- Download a Live CD version of Ubuntu Linux and boot the server with the CD. This way you won't need a functioning hard disk for the server to keep working.
- Restore your backup to another server with enough disk space to hold your web and database data.
- NFS mount the backups to the server that is now running off of the live CD.
- symlink the web root, database data directory, and apache configuration files into where they were originally located
- restart the web and database servers
- cross your fingers and hope that it will work

The details:

Ubuntu

- I'm going to assume that [getting](#) and booting Ubuntu is not a problem... the standard download should be a live CD version
- General notes:
 - the default password for the automatically logged on "ubuntu" user is blank
 - to switch to root, type this: "sudo -s" ... the password is the same as the password for the ubuntu user
 - to install packages, use Synaptic (on the menu: System ? Administration ? Synaptic) or use apt-get from the shell:

```
apt-get install package_name
```

- After you boot into Ubuntu, there are a few things you need to do:
 - change the user password (I'm not sure if this is necessary, but a user with a blank password makes me nervous): to do this, just open a shell session and run passwd
 - Change the network configuration to match your old server. There's a graphical tool for this in the menu: System ? Administration ? Networking
 - change the IP address information, DNS settings, etc. ... also change the hostname
 - Log off and log back on... but don't reboot! (this seems to be necessary after changing the hostname or strange things happen)

NFS

- You'll need to install a few extra packages on the live CD server: (also install whatever dependencies come up)
 - For NFS:
 - nfs-common
 - portmap
 - For MySQL, PHP, and Apache:
 - mysql-common
 - mysql-client-5.0
 - mysql-server-5.0
 - php5-mysql
 - (note that this also installs PHP and Apache because they get pulled in as dependencies)
- Setup a NFS export on the spare server (this is where you restored your backup to)
 - If you don't have a NFS server installed, here are the packages you need for Debian:
 - nfs-kernel-server
 - nfs-common
 - portmap
 - Server configuration:
 - Add an export to /etc/exports:

```
/path/to/restored/backup 123.123.123.123(rw,async,no_root_squash,insecure_locks)
```

- change the IP address to the address of the live CD server
- this configuration is fairly insecure... but it's restricted to just one server, and this setup should only be temporary while you get a replacement drive
- reload the exports with this command:

```
exportfs -a
```

- Setup the client side of the NFS export (this is on the live CD server)

- make a directory to mount the restored data to:

```
mkdir /mnt/restore
```

- edit your /etc/fstab file and add this:

```
spare_server_hostname:/path/to/restored/backup /mnt/restore nfs rw,rsize=4096,wsiz=4096,
```

- mount it:

```
mount /mnt/restore
```

Server and Filesystem Shenanigans

- The following is all done on the live CD server
- Stop the web and database servers:

```
/etc/init.d/mysql stop  
/etc/init.d/apache2 stop
```

- Make symlinks:
 - make a symbolic link to wherever your web root was originally located (erase any existing directory first)
 - symlink to your database data directory:

```
cd /var/lib  
rm -Rf mysql  
ln -s /mnt/restore/var/lib/mysql
```

- fix the debian mysql maintenance account password:

```
cd /etc/mysql
rm debian.cnf
ln -s /mnt/restore/etc/mysql/debian.cnf
```

- symlink to your original apache configuration:

```
cd /etc
rm -Rf apache2
ln -s /mnt/restore/etc/apache2
```

- Fix any ownership weirdness:

- The user ID's on the live CD and the original server may not all match up, so we need to make the ID's match up with the live CD:

```
chown -R www-data.www-data /mnt/restore/path/to/webroot
chown -R mysql.mysql /mnt/restore/var/lib/mysql
```

- note: don't chown the symlinks because it'll only change the symlink (and not the underlying files)

- The big moment...:

- start up your servers:

```
/etc/init.d/mysql start
/etc/init.d/apache2 start
```

- if you're lucky, you'll have no error messages
- Apache might give some errors for missing modules. You can either install the missing modules (using apt-get or synaptic) or disable the modules if they're not used:

```
a2dismod name_of_module
```

- Now go check that your site is actually working properly. If it's not working... figure out what's wrong and add your solutions to this article. (please! :-)

Revision #10

Created 2006-08-15 20:04:52 UTC by Fong, Jason

Updated 2011-05-10 22:06:15 UTC by Franks, Mike