

# Mise en place d'un serveur LAP



## Table des matières

- Introduction : ..... 2
- Installation des services : ..... 2
- Test serveur Web : ..... 3
- C'est quoi un Vhost ? ..... 4
  - Configuration d'un Vhost : ..... 4
- Créer un fichier test html : ..... 7
- Test du Vhost avec un ordinateur client : ..... 8
- Comment télécharger un logicielle sur un Vhost : ..... 9
- Conclusion : ..... 10

## Introduction :

Dans le cadre pédagogique, nous avons à notre disposition un serveur de virtualisation (PVE). Sur ce, serveur de virtualisation, il s'y trouve :

- Un réseau local, adresser en 192.168.200.0 /24
- Une passerelle adressée en 192.168.200.254 /24
- Un client DHCP

LAMP est un acronyme

- **L**inux : Le système d'exploitation
- **A**pache : Le serveur Web
- **M**ySQL / MariaDB : Le serveur de Base de données
- **P**HP : Le langage de script (Hypertext Preprocessor)

Nous n'utiliserons pas MySQL, nous utiliserons l'acronyme LAP, donc Apache et PHP, Linux est juste le système d'exploitation.

PHP est un langage de scripts généraliser et Open Source, spécialement conçu pour développement d'application web.

Le PHP s'intègre facilement au HTML, le grand avantage du PHP est qu'il est extrêmement simple. Apache http Serveur est serveur http open-source pour les systèmes d'exploitation modernes, y compris UNIX et Windows. Le but est de fournir un serveur sécurisé, efficace et extensible qui fournit des services HTTP standard et HTTPS sécuriser. Apache vient d'un projet d'Apache Software Foundation.

À la fin nous aurons un réseau local avec un serveur Web en interne

## Installation des services :

Pour installer Apache et PHP il suffit d'utiliser cette commande pour les installés tous les deux, **apt install apache2 PHP**, pour PHP il installera la dernière version stable. Pour savoir si le service est en fonction il suffit de faire cette commande, **systemctl status apache2.services** pour savoir si PHP est bien activé sur apache2 il vous suffit de vérifier la version de PHP installer **PHP -v** ce qui vous donnera la version, puis effectuer cette commande, **a2enmode php7.3 (7.3 versions installer)** ceci vous dira si PHP est actif sur apache.

Des modules complémentaires son disponible sur PHP, vous pouvez les installer comme ceci **Apt install PHP-{curl,gd,intl,memcache,xml,zip,mbstring,json}** ceci sont les modules les plus courants.

- Curl = permet de communiquer facilement avec de nombreux types de serveurs applicatifs en parlant le même langage.
- Gd = gif draw servant à manipuler des images dynamiques.
- Intl = internationalisation ceci permet de présenter votre cite dans d'autres langues et cultures.
- Memcahce = Serveur de cache, il agit comme un démon UNIX il enregistre tout type de variable.
- Xml = ceci permet de lire les documents XML, ce qui traduit la langue des balises extensible. Exemple un fichier Excel
- Zip = pour pouvoir compresser où décompresser des fichiers.
- Mbstring = fonction d'expression régulière avec le support de caractère multioctets.
- Json = JavaScript Objet Notation permet de lire les objets JavaScript.

Une fois installer, un test simple et possible, il suffit de créer un fichier, cette commande permet d'afficher de nombreuses informations sur PHP concernant sa configuration : les options de compilation, extensions, version, information sur le serveur, et l'environnement PHP, les informations sur le système, les chemins, valeurs générales et locales de configuration, les en-têtes http et la licence PHP. `Phpinfo()` est un outil de débogage, car il affiche le contenu de toutes les variables (Environnement, GET, POST, les Cookies, et le Serveur) EGPCS. Nous pouvons aussi personnaliser l'affichage avec plusieurs paramètres. Voici des options de `phpinfo()`.

Nom de la constante	Valeur	Description
INFO_GENERAL	1	La ligne de configuration, le chemin du <code>php.ini</code> , la date de compilation, le serveur web, le système, etc.
INFO_CREDITS	2	Les crédits de PHP. Voir aussi <a href="#">phpcredits()</a> .
INFO_CONFIGURATION	4	Valeurs courantes locales et générales des directives PHP. Voyez aussi la fonction <code>ini_get()</code> .
INFO_MODULES	8	Modules chargés et leur configuration spécifique. Voir aussi la fonction <code>get_loaded_extensions()</code> .
INFO_ENVIRONMENT	16	Informations sur les variables d'environnement, qui sont disponibles dans la variable <code>\$_ENV</code> .
INFO_VARIABLES	32	Affiche toutes les <u>variables prédéfinies</u> , issues de l'environnement, la méthode GET, la méthode POST, les cookies et le serveur.
INFO_LICENSE	64	La licence PHP. Voir aussi » <a href="#">la FAQ de la licence</a> .
INFO_ALL	-1	Affiche toutes les informations suscitées.

Voici la commande qui nous permet d'afficher tous les informations, mais ATTENTION ne pas laisser cette page sur le serveur, il faut bien là supprimer après avoir fait des tests, ceci ne doit pas être laissé en production :

`echo '<?php phpinfo(); ?>' >/var/www/html/info.php` si cette page s'affiche elle nous permettra de conclure que le service PHP fonctionne.

Toutes les pages Web se trouvent dans le dossier HTML, vous pouvez vous y rendre avec cette commande `cd /var/www/html` vous trouverez la page `info.php` et une page `index.html`, cette page nous permet de vérifier si apache fonctionne, il y a aussi des instructions qui permet de maîtriser apache.

## Test serveur Web :

Pour tester le serveur web nous allons utiliser notre client, pour afficher les pages web nous utiliserons l'adresse IP fixe du serveur suivi du fichier que vous voulez consulter.

The screenshot shows a web browser window with the address `192.168.200.253`. The page displays the Apache2 Debian Default Page with the message "It works!". Below this, there is a "Configuration Overview" section. To the right of the browser window, a PHP info page is visible, showing the following details:

PHP Version 7.3.19-1-deb10u1	
System	Linux LAP 4.19.0-13-amd64 #1 SMP Debian 4.19.160-2 (2020-11-28) x86_64
Build Date	Jul 5 2020 06:46:45
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.3/apache2
Loaded Configuration File	/etc/php/7.3/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.3/apache2/conf.d
Additional .ini files parsed	/etc/php/7.3/apache2/conf.d/10-mysqld.ini, /etc/php/7.3/apache2/conf.d/10-opcache.ini, /etc/php/7.3/apache2/conf.d/10-pdo.ini, /etc/php/7.3/apache2/conf.d/15-xml.ini, /etc/php/7.3/apache2/conf.d/20-bcmath.ini, /etc/php/7.3/apache2/conf.d/20-bz2.ini, /etc/php/7.3/apache2/conf.d/20-calendar.ini, /etc/php/7.3/apache2/conf.d/20-ctype.ini, /etc/php/7.3/apache2/conf.d/20-curl.ini, /etc/php/7.3/apache2/conf.d/20-dom.ini, /etc/php/7.3/apache2/conf.d/20-exif.ini, /etc/php/7.3/apache2/conf.d/20-fileinfo.ini, /etc/php/7.3/apache2/conf.d/20-ftp.ini, /etc/php/7.3/apache2/conf.d/20-gd.ini, /etc/php/7.3/apache2/conf.d/20-gettext.ini, /etc/php/7.3/apache2/conf.d/20-iconv.ini, /etc/php/7.3/apache2/conf.d/20-imageick.ini, /etc/php/7.3/apache2/conf.d/20-imap.ini, /etc/php/7.3/apache2/conf.d/20-intl.ini, /etc/php/7.3/apache2/conf.d/20-json.ini, /etc/php/7.3/apache2/conf.d/20-mbstring.ini, /etc/php/7.3/apache2/conf.d/20-mysqli.ini, /etc/php/7.3/apache2/conf.d/20-pdo_mysql.ini, /etc/php/7.3/apache2/conf.d/20-phar.ini, /etc/php/7.3/apache2/conf.d/20-posix.ini, /etc/php/7.3/apache2/conf.d/20-readline.ini, /etc/php/7.3/apache2/conf.d/20-shmop.ini, /etc/php/7.3/apache2/conf.d/20-simplexml.ini, /etc/php/7.3/apache2/conf.d/20-soap.ini, /etc/php/7.3/apache2/conf.d/20-sockets.ini, /etc/php/7.3/apache2/conf.d/20-sysmsg.ini, /etc/php/7.3/apache2/conf.d/20-sysvsem.ini, /etc/php/7.3/apache2/conf.d/20-sysvshm.ini, /etc/php/7.3/apache2/conf.d/20-tidy.ini, /etc/php/7.3/apache2/conf.d/20-tokenizer.ini, /etc/php/7.3/apache2/conf.d/20-wddx.ini, /etc/php/7.3/apache2/conf.d/20-xmlreader.ini, /etc/php/7.3/apache2/conf.d/20-xmlrpc.ini, /etc/php/7.3/apache2/conf.d/20-xmlwriter.ini, /etc/php/7.3/apache2/conf.d/20-xsl.ini, /etc/php/7.3/apache2/conf.d/20-zip.ini
PHP API	20180731
PHP Extension	20180731
Zend Extension	320180731
Zend Extension Build	API320180731.NTS

## C'est quoi un Vhost ?

Les Vhost qui viennent de l'anglais virtual hosting, est une méthode que le serveur web utilise pour accueillir plusieurs noms de domaine sur le même serveur et la même IP tout en maintenant une gestion des noms séparément.

Par exemple si nous utilisons Nextcloud un hébergeur de fichier, si nous voulons l'installer sur notre serveur web en local, il nous suffira de créer et de configurer un vhost Nextcloud, très important il faut aussi le déclarer sur le serveur DNS internet, donc si nous le déclarons, pour afficher notre Vhost il faudra rechercher <http://nextcloud.pmmonnier.hyp> Attention cette recherche sera effectuée en interne, non sécuriser le trafic web sera en clair sur le réseau. Pour le configurer il faut activer le mode SSL sur apache (Transport Layer Security) ce qui activera le mode HTTPS avec le protocole 443.

Configuration d'un Vhost :

Pour configurer un vhost il faut activer le mode SSL **a2enmod ssl**, pour que le trafic circule sur le réseau soit chiffré, ceci permet d'éviter qu'une personne mal intentionnée analyse le réseau et récupère des informations. Effectuer un redémarrage du service apache2 **systemctl restart apache2.service**

Pour créer un Vhost, il faut se rendre dans les fichiers de configuration de apache2 **/etc/apache2**. Pour afficher les fichiers se trouvant dans le répertoire apache2 il suffit de taper **ls** (l'abréviation de liste en anglais) ceci permet de lister sont contenu.

Pour trouver le fichier de configuration il faut se rendre dans le dossier **sites-available** (sites disponibles)

```
root@LAP:~# cd /etc/apache2/
root@LAP:/etc/apache2# ls
apache2.conf  conf-enabled  magic          mods-enabled  sites-available
conf-available  envvars      mods-available  ports.conf    sites-enabled
root@LAP:/etc/apache2# cd sites-available/
root@LAP:/etc/apache2/sites-available# ls
000-default.conf  default-ssl.conf
root@LAP:/etc/apache2/sites-available#
```

Dans le fichier sites-available, nous trouvons les fichiers de configuration, celui qui nous intéresse et le **default-ssl.conf**, avant toute modification, il faut le copier et le renommer, **cp default-ssl.conf nextcloud\_vhost-ssl.conf** puis le configurer, **nano nextcloud\_vhost-ssl.conf**

```
GNU nano 3.2 nextcloud_vhost-ssl.conf
1 <IfModule mod_ssl.c>
2   <VirtualHost _default_:443>
3     ServerAdmin webmaster@localhost
4
5     DocumentRoot /var/www/html
6
7     # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
8     # error, crit, alert, emerg.
9     # It is also possible to configure the loglevel for particular
10    # modules, e.g.
11    #LogLevel info ssl:warn
12
13    ErrorLog ${APACHE_LOG_DIR}/error.log
14    CustomLog ${APACHE_LOG_DIR}/access.log combined
15
16    # For most configuration files from conf-available/, which are
17    # enabled or disabled at a global level, it is possible to
18    # include a line for only one particular virtual host. For example the
19    # following line enables the CGI configuration for this host only
20    # after it has been globally disabled with "a2disconf".
21    #Include conf-available/serve-cgi-bin.conf
22
23    # SSL Engine Switch:
24    # Enable/Disable SSL for this virtual host.
25    SSLEngine on
26
27    # A self-signed (snakeoil) certificate can be created by installing
28    # the ssl-cert package. See
29    # /usr/share/doc/apache2/README.Debian.gz for more info.
30    # If both key and certificate are stored in the same file, only the
31    # SSLCertificateFile directive is needed.
32    SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
33    SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
```

Dans ce fichier nous avons :

1 <sup>ère</sup> ligne : le mode (si nous sommes sécurisés ou pas d'SSL)
2 <sup>ème</sup> ligne : le type de protocole est utilisé 443 (https)
5 <sup>ème</sup> ligne : où se trouvent le ou les fichiers du vhost
32 <sup>ème</sup> ligne : où se trouvent les certificats (clef publique)
33 <sup>ème</sup> ligne : où se trouvent les certificats (clef priver)

Voici les modifications à apporter au fichier.

```

1 <IfModule mod_ssl.c>
2 <VirtualHost *:80>
3 ServerName nextcloud.pmmonnier.hyp
4 RedirectPermanent / "https://nextcloud.pmmonnier.hyp/"
5 </VirtualHost>
6
7 <VirtualHost _default_:443>
8     ServerAdmin webmaster@nextcloud.pmmonnier.hyp
9     ServerName nextcloud.pmmonnier.hyp
10    DocumentRoot /var/www/html/nextcloud
11
12    # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
13    # error, crit, alert, emerg.
14    # It is also possible to configure the loglevel for particular
15    # modules, e.g.
16    #LogLevel info ssl:warn
17
18    ErrorLog ${APACHE_LOG_DIR}/error.log
19    CustomLog ${APACHE_LOG_DIR}/access.log combined
20
21    # For most configuration files from conf-available/, which are
22    # enabled or disabled at a global level, it is possible to
23    # include a line for only one particular virtual host. For example the
24    # following line enables the CGI configuration for this host only
25    # after it has been globally disabled with "a2disconf".
26    #Include conf-available/serve-cgi-bin.conf
27
28    # SSL Engine Switch:
29    # Enable/Disable SSL for this virtual host.
30    SSLEngine on
31
32    # A self-signed (snakeoil) certificate can be created by installing
33    # the ssl-cert package. See
34    # /usr/share/doc/apache2/README.Debian.gz for more info.
35    # If both key and certificate are stored in the same file, only the
36    # SSLCertificateFile directive is needed.
37    SSLCertificateFile /etc/apache2/ssl/nextcloud-selfsigned.crt
38    SSLCertificateKeyFile /etc/apache2/ssl/nextcloud-selfsigned.key

```

1 <sup>ère</sup> ligne : le module d'apache pour le https
2 <sup>ème</sup> ligne : le VirtualHost du port 80 (http) qui n'est pas sécurisé
3 <sup>ème</sup> ligne : ServerName qui est le nom de domaine du vhost
4 <sup>ème</sup> ligne : RedirectPermanent qui indique aux vhost que si une requête du protocole 80 et effectuée, alors il redirige la personne sur cette URL qui est une bascule sur le Port 443 qui lui est sécuriser.
7 <sup>ème</sup> ligne : Qui appartient se serveur, donc quel compte administre se site WEB
8 <sup>ème</sup> ligne : quelle est son FQDN (Fully Qualified Domain Name) nom de domaine complètement qualifié.
36 <sup>ème</sup> ligne : le fichier de certificat SSL avec comme convention le nom du certificat.
37 <sup>ème</sup> ligne nous avons la même chose bien évidemment ces 2 fichiers ne contient pas les mêmes informations.

Création des certificats auto signer :

Nous créons des certificats auto signer car nous avons l'usage du protocole HTTPS en interne, ce qui fait que nous pouvons en créer autant que l'on veut, car il sera automatiquement signé. Si je voudrais avoir un certificat signer pour héberger un site WEB en externe, il faut qu'il soit délivré par un organisme appelé autorité de certification, notée CA (Certification Authority) ils sont chargés de délivrer les certificats, de leur assigner une date de validité (date limite d'utilisation) et d'annuler

éventuellement des certificats avant cette date en cas d'usurpation de la clé ou du propriétaire. Dans notre cas nous utiliserons le certificat serveur, il permet de garantir que l'URL et en particulier le domaine de la page web appartient bien à telle ou telle entreprise. Il permet aussi de sécuriser les transactions avec les utilisateurs grâce au protocole SSL.

Pour créer un certificat, j'ai délocalisé le fichier SSL, pour tout centraliser. **mkdir /etc/apache2/ssl** puis utiliser cette commande pour générer un certificat auto signer. **openssl req -x509 -sha256 -nodes -days 730 -newkey rsa:2048 -keyout /etc/apache2/ssl/nextcloud-selfsigned.key -out /etc/apache2/ssl/nextcloud-selfsigned.crt**

X509	Ceci est la structure des certificats, car elle est normalisée (X.509v3) ce qui correspond à la version du certificat
Sha256	Secure Hash Algorithm ceci permet de sécuriser le site internet
Days 730	Le nombre de jours où sera valide le certificat (2ans)
Newkey rsa :2048	Le chiffrement RSA est l'algorithme de cryptographie
-keyout	Là où la clé sera enregistrée
nextcloud-selfsigned.key	Nextcloud = le nom de votre fichier selfsigned.key = le type de fichier (clé).
-out	Renvoie le certificat dans le dossier de destination

```

root@LAP:/etc/apache2/sites-available# openssl req -x509 -sha256 -nodes -days 730 -newkey rsa:2048 -keyout /etc/apache2/ssl/nextcloud-selfsigned.key -out /etc/apache2/ssl/nextcloud-selfsigned.crt
Generating a RSA private key
.....+++++
.....+++++
writing new private key to '/etc/apache2/ssl/nextcloud-selfsigned.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:Creuse
Locality Name (eg, city) []:Aubusson
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Lycée Eugène Jamot
Organizational Unit Name (eg, section) []:BTS SIO
Common Name (e.g. server FQDN or YOUR name) []:nextcloud.pmmonnier.hyp
Email Address []:contact@pmmonnier.ovh
root@LAP:/etc/apache2/sites-available# _

```

Une fois la commande tapée il nous demande des informations :

Country Name	Nom du pays = FR
State Or Province Name	Nom de l'état ou de la province = Creuse
Locality Name	Nom de la localité = Aubusson
Organization Name	Nom de l'organisation = Lycée Eugène Jamot
Organisation Unit Name	Nom de l'unité de l'organisation = BTS SIO
Common Name FQDN	Nom de domaine complètement qualifié.
Email Address	L'adresse mail = contact@pmmonnier.ovh

Une fois le certificat terminer il faut activer le Vhost, pour ce faire **a2ensite nextcloud\_vhost-ssl.conf** puis redémarrer le service apache2

## Créer un fichier teste html :

Voici le code html de base, ceci nous permettra d'afficher une page depuis notre Vhost. Pour se faire se rendre dans les dossiers **/var/www/html/nextcloud** créer un fichier `index.html` dans le répertoire de nextcloud, utiliser ce code pour tester votre Vhost. Puis redémarre apache2 pour prendre les modifications en compte.

```
root@LAP:/var/www/html/nextcloud# cat index.html
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Nextcloud</title>
  </head>
  <body>
    <p>Bonjour test</p>
  </body>
</html>
root@LAP:/var/www/html/nextcloud# _
```

Pour que la résolution Nom fonctionne, il ne faut pas oublier de le renseigner dans le serveur DNS.

```
;
; BIND data file for pmmonnier.hyp interface
;
$TTL      604800
@         IN      SOA     ns1.pmmonnier.hyp. root.pmmonnier.hyp. (
                        2           ; Serial
                        604800      ; Refresh
                        86400       ; Retry
                        2419200     ; Expire
                        604800 )    ; Negative Cache TTL
;
@         IN      NS     ns1.pmmonnier.hyp.
ns1       IN      A      192.168.200.252
LAP       IN      A      192.168.200.253
BDD       IN      A      192.168.200.251
nextcloud IN      CNAME   LAP
```

Ceci se trouve dans le fichier d'enregistrement DNS **/etc/bind/db.pmmonnier.hyp** , il faut déclarer le vhost **nextcloud** en tant que CNAME puis la redirection se fera sur le LAP, car c'est lui qui stocke le Vhost nextcloud.

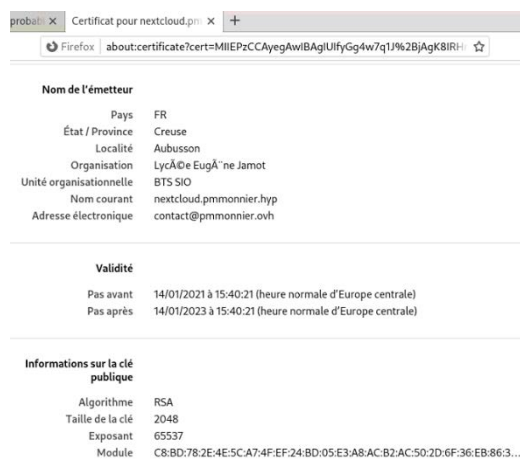
## Test du Vhost avec un ordinateur client :

Avant de tester la page web, nous utiliserons la commande dig pour effectuer des requête DNS, dans notre cas je vais interroger le serveur DNS interne, qui est **nextcloud.pmmonnier.hyp** il va pouvoir me dire s'il me renvoie bien sur le serveur LAP.

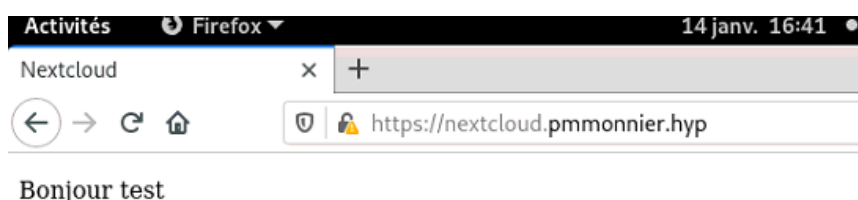
```
Pierrick@fedora:~  
[Pierrick@localhost ~]$ dig nextcloud.pmmonnier.hyp  
; <<> DiG 9.11.25-RedHat-9.11.25-2.fc33 <<> nextcloud.pmmonnier.hyp  
;; global options: +cmd  
;; Got answer:  
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 13792  
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1  
  
;; OPT PSEUDOSECTION:  
; EDNS: version: 0, flags:; udp: 65494  
;; QUESTION SECTION:  
;nextcloud.pmmonnier.hyp.      IN      A  
  
;; ANSWER SECTION:  
nextcloud.pmmonnier.hyp. 604800 IN      CNAME  LAP.pmmonnier.hyp.  
LAP.pmmonnier.hyp.      7199   IN      A      192.168.200.253  
  
;; Query time: 3 msec  
;; SERVER: 127.0.0.53#53(127.0.0.53)  
;; WHEN: jeu. janv. 14 16:10:47 CET 2021  
;; MSG SIZE rcvd: 86  
  
[Pierrick@localhost ~]$
```

La commande dig me renvoie bien sûr le **LAP.pmmonnier.hyp** avec l'IP du serveur LAP.

Test de la page web, il nous faut un client Web, donc Mozilla Firefox, nous allons faire la recherche **nextcloud.pmmonnier.hyp** comme nous avons la redirection du protocole http port 80 il nous redirige automatiquement sur le protocole https port 443, nous avons notre certificat autosigné qui est en fonction, le client web nous avertie qu'il y a un risque potentiel de sécurité, car il est autosigné.



Si nous acceptons les risques, nous pouvons remarquer que notre vhost fonctionne.





## Comment télécharger un logiciel sur un Vhost :

Pour installer un logiciel, comme nextcloud pour une gestion de partage de fichier, il est possible de l'installer sur un serveur, comme nous procédons un vhost nextcloud, nous allons le télécharger.

Pour ce faire nous devons utiliser la commande wget qui permet de récupérer un fichier depuis internet, wget suivie de l'adresse html. Avant, téléchargé, il faut se placer dans le dossier temporaire.

```
cd /tmp
```

```
Wget https://download.nextcloud.com/server/releases/nextcloud-16.0.1.zip
```

Le fichier sera compressé en point ZIP il suffit d'utiliser unzip suivie du nom du fichier.

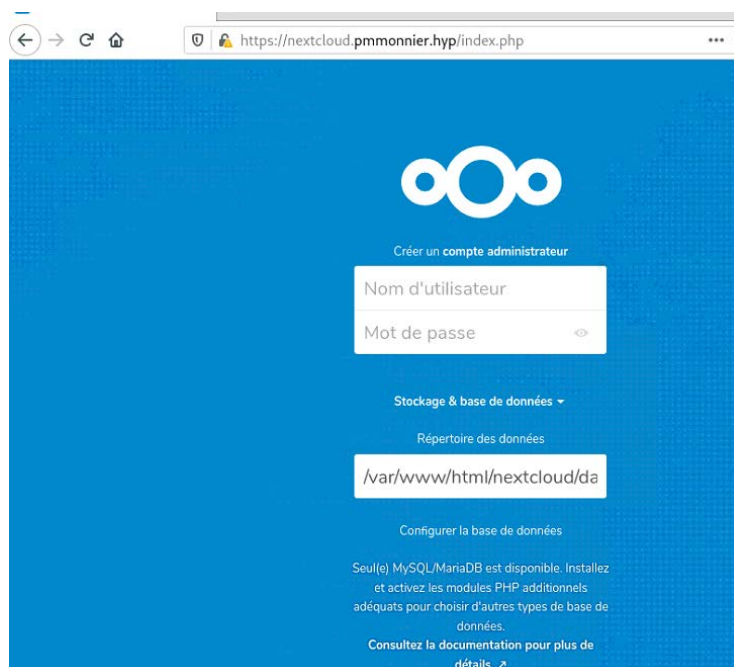
```
Déplacer le fichier dans l'arborescence /var/www/html
```

```
mv nextcloud /var/www/html
```

Pour qu'apache puisse lire et écrire, il lui faut les droits sur le dossier. Nous devons nous situer dans ces dossiers /var/www/html puis faire un **chown -R www-data :www-data nextcloud**

**www-data : www-data** sont les droits d'utilisateur d'apache, si nous ne lui donnons pas les droits il y aura des erreurs l'or d'une installation.

```
index.html index.php nextcloud
root@LAP:/var/www/html# chown -R www-data:www-data nextcloud/
root@LAP:/var/www/html# ls
index.html index.php nextcloud
root@LAP:/var/www/html# ls -l
total 20
-rw-r--r--  1 root    root    10701 janv.  5 21:43 index.html
-rw-r--r--  1 root    root         20 janv.  5 22:23 index.php
drwxr-xr-x 14 www-data www-data 4096 mai  15 2019 nextcloud
root@LAP:/var/www/html# _
```



Test du Vhost nextcloud depuis notre machine cliente, ce que nous pouvons constater c'est que notre téléchargement nextcloud sur notre vhost à fonctionner, il nous affiche la page d'administration et de configuration.

## Conclusion :

Nous pouvons constater que les pages web se sont bien affichées, que le vhost est bien sécurisé en https, donc notre serveur web fonctionne.