

Serveur web

Mise en place et sécurisation d'un serveur

Yoann Coualan

Qu'est ce qu'un serveur web ?

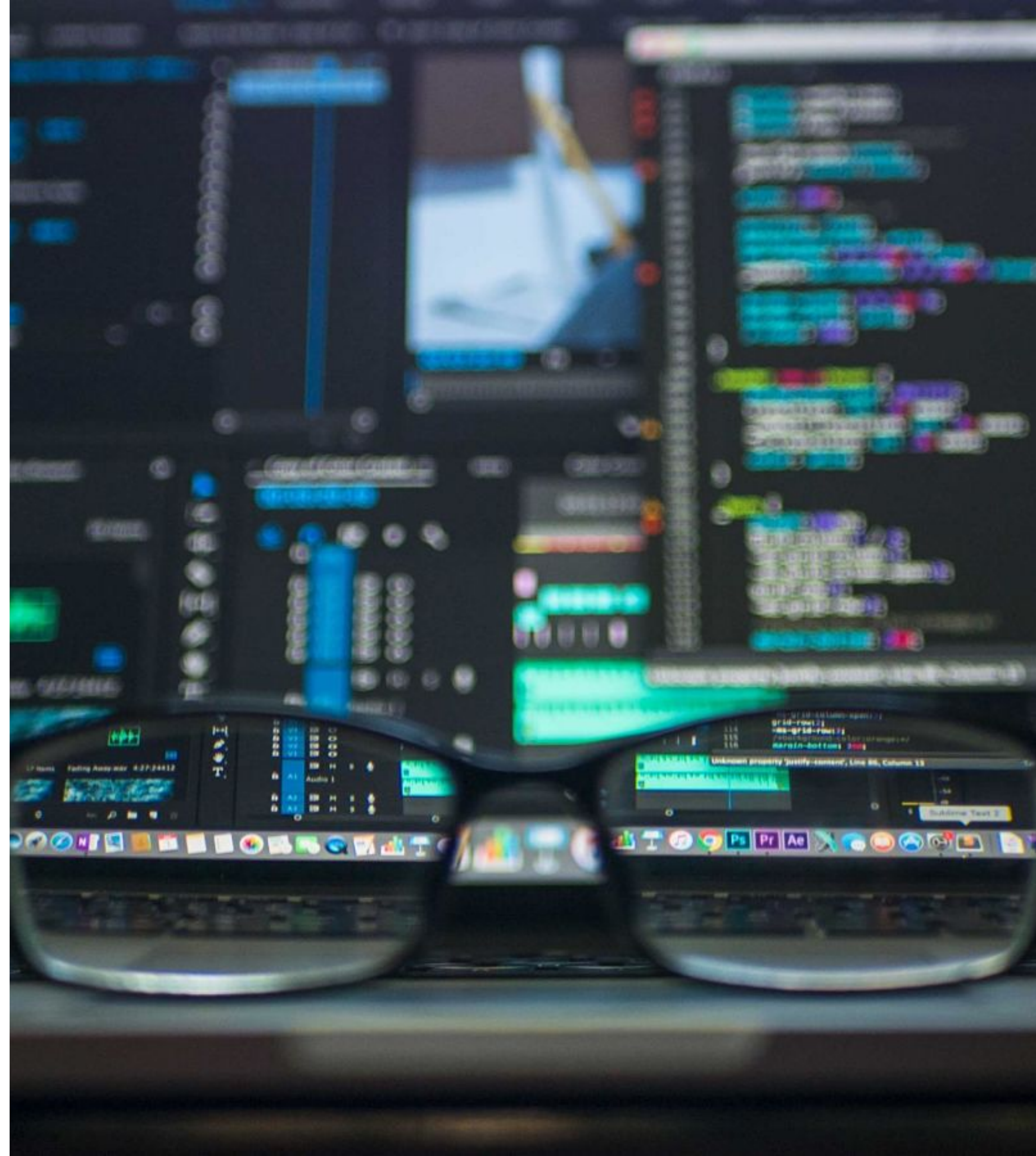
Un ensemble de **composants matériels et/ou logiciels** qui fonctionnent ensemble.

Peut prendre une forme physique ou virtuelle.



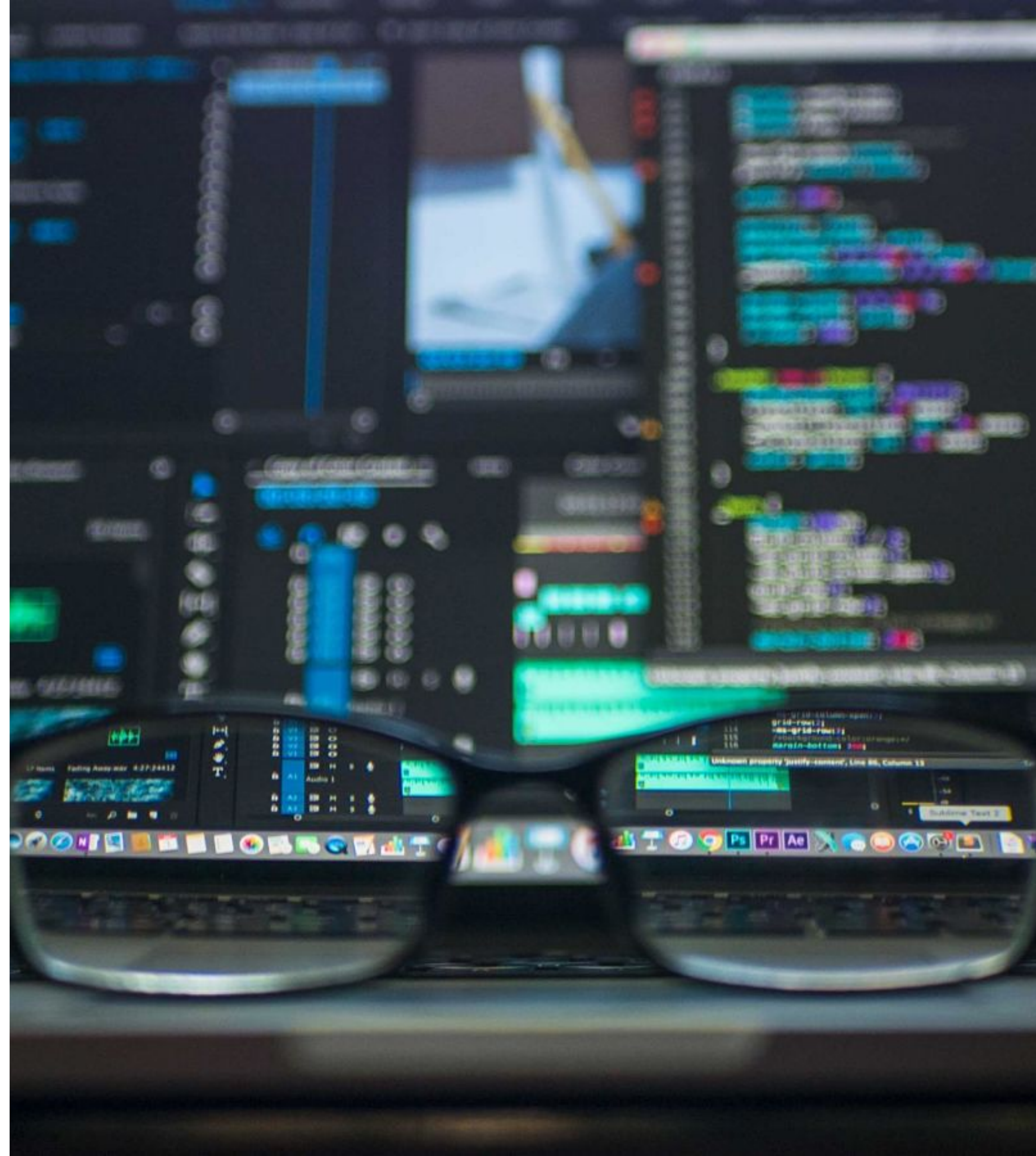
VPS

- **Virtual Private Server**
- Partitionnement d'un serveur classique
- Réduction des coûts
- Conserve les avantages d'un serveur dédié physique



SSH

- Secure Shell
- Protocole permettant d'établir une connexion chiffrée entre votre ordinateur et un serveur



Connexion au serveur

Connexion :

```
ssh [utilisateur]@[IP]
```

Passage en super-admin

```
sudo su
```

Ajouter un utilisateur

Création de l'utilisateur

```
adduser user_name
```

Voir les groupes de l'utilisateur

```
groups user_name
```

Ajouter un utilisateur au group sudo

(-aG ajoute et ne remplace pas la liste existante)

```
usermod -aG sudo user_name
```


Editer l'utilisateur Debian

Changer le mot de passe

```
passwd
```

Déconnexion, puis connexion avec un autre utilisateur sudo

Supprimer l'accès sudo de Debian

```
gpasswd -d debian sudo
```

Serveur HTTP : Apache VS Nginx

APACHE	NGINX
Serveur web Proxy	Serveur web Proxy Load-balancer

Installation de Nginx

```
apt-get install nginx
```

Maintenant votre URL est accessible

Configurer Nginx

Répertoire de configuration :

```
/etc/nginx/conf.d/
```

Créer / éditer un fichier de configuration :

```
nano /etc/nginx/conf.d/default.conf
```

Sauvegarder un fichier

```
CTRL + X
```


Configurer Nginx - HTML

```
server{  
    listen 443 ssl;  
    server_name votre-nom-de-domaine.fr;  
  
    access_log /chemin/vers/fichier/log.log;  
    error_log /chemin/vers/fichier/erreurs.log;  
  
    root /chemin/vers/repo;  
    index index.html;  
  
    location / {  
        try_files $uri $uri/ =404;  
    }  
}
```

Configurer Nginx - HTML

Sur **CPanel**, il suffit de sélectionner le bon répertoire cible dans la configuration du domaine ou sous-domaine pour que cela fonctionne.

Configurer Nginx

Tester la configuration de nginx

```
nginx -t
```

Redémarrer nginx

```
service nginx restart
```

Création d'un fichier HTML

Création du dossier "default" dans "/home" :

```
mkdir /home/default
```

Création d'un fichier HTML dans "/home/default" :

```
nano /home/default/index.html
```


Certbot

Installe les certificats SSL pour passer en HTTPS



Certbot

Installer snapd

```
apt install snapd
```

Installation de core de snap pour avoir la dernière version

```
snap install core; snap refresh core
```

Installation de certbot

```
snap install --classic certbot
```

Création d'un lien symbolique certbot

```
ln -s /snap/bin/certbot /usr/bin/certbot
```


Certbot

Autoriser la vérification de certificat dans nginx

```
#/etc/nginx/conf.d/letsencrypt.conf
server {
    # .well-known doit resté accessible
    location ~ /\.well-known/acme-challenge {
        allow all;
    }
    # On interdit habituellement l'accès au dotfiles
    location ~ /\. { deny all; access_log off; log_not_found off; }
}
```

Création d'un certificat

```
certbot --nginx
```


PHP-FPM

FastCGI Process Manager

Permet la communication entre un serveur web et PHP



Installation de PHP-FPM

Vérification que le paquet ne soit pas déjà installé :

```
apt-cache search php-fpm
```

PHP-FPM est déjà installé sur notre distribution (Debian)

Installation de PHP8.2

Ajout d'un repo possédant PHP 8.2 :

```
apt install lsb-release apt-transport-https ca-  
certificates software-properties-common
```

Import de la signature :

```
wget -O /etc/apt/trusted.gpg.d/php.gpg https://  
packages.sury.org/php/apt.gpg
```

Ajout du repo à la liste des paquets connus :

```
sh -c 'echo "deb https://packages.sury.org/php/ $  
(lsb_release -sc) main" > /etc/apt/sources.list.d/  
php.list'
```

Installation de PHP8.2

Mise à jour des repos

```
apt update
```

Installation de PHP8.2

```
apt install php8.2
```

Vérification de la version PHP

```
php -v
```

Installation de PHP8.2

Installer des extensions PHP8

```
apt install php8.2-  
{cli,zip,mysql,bz2,curl,mbstring,intl,common,xml,dom}
```

Ne pas oublier d'installer FPM

```
apt install php8.2-fpm
```


Configurer Nginx - PHP

```
server{  
  
    listen 443 ssl;  
  
    server_name votre-nom-de-domaine.fr;  
  
  
  
    access_log /chemin/vers/fichier/log.log;  
    error_log /chemin/vers/fichier/erreurs.log;  
  
  
  
    root /chemin/vers/repo;  
  
    index index.html;  
  
  
  
    location ~ \.php$ {  
        try_files $uri =404;  
  
        fastcgi_pass unix:/var/run/php/php8.3-fpm.sock;  
  
        # Si aucun nom de fichier dans l'URL, on change index.php  
  
        fastcgi_index index.php;  
  
        # Envois à SCRIPT_FILENAME de fastcgi le fichier appelé  
  
        fastcgi_param SCRIPT_FILENAME $document_root$fastcgi_script_name;  
  
        # Inclus d'autres paramètres nécessaires au fonctionnement de fastcgi  
  
        include fastcgi_params;  
    }  
}
```

Création d'un fichier PHP

Création du dossier "php" dans "/home" :

```
mkdir /home/php
```

Création d'un fichier PHP dans "/home/php" :

```
nano /home/php/index.php
```

Avec CPanel, là aussi, il suffit de changer la destination du sous-domaine.

Afficher le contenu d'un fichier

Editer le fichier

```
nano /chemin/vers/fichier
```

Afficher les dernières lignes

```
tail -f /chemin/vers/fichier
```


MariaDB

Installation du paquet MariaDB

```
apt install mariadb-server
```

Restreindre l'accès à MySQL

```
mysql_secure_installation
```

Connexion à MySQL

```
mysql -u root -p
```

MariaDB - bases de données

Afficher les bases de données

```
show databases;
```

Créer une base de données

```
create database [db_name];
```

MariaDB - utilisateurs

Création d'un utilisateur

```
CREATE USER 'new_user'@'localhost' IDENTIFIED BY 'password';
```

Définir les rôles

```
GRANT ALL PRIVILEGES ON db_name.* TO 'new_user'@'localhost';
```

Appliquer les rôles

```
FLUSH PRIVILEGES;
```


Installer GIT

Installation de git

```
apt-get install git
```

Récupérer son token github

Github → Settings → Developer Settings → Personal Access

Token → Generate New Token → Remplir le formulaire → Generate

token → Copier et sauvegarder le Token généré

Installer Oh-My-Zsh

Installer ZSH

```
apt install zsh
```

Installer Oh-My-Zsh

```
sh -c "$(curl -fsSL https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"
```

```
server {  
    listen 443 ssl;  
  
    server_name votre-nom-de-domaine.fr;  
  
    access_log /chemin/vers/fichier/log.log;  
  
    error_log /chemin/vers/fichier/erreurs.log;  
  
    root /repo/symfony/public;  
  
    client_max_body_size 0;  
  
    fastcgi_read_timeout 300;  
  
    location / {  
        try_files $uri /index.php$is_args$args;  
    }  
  
    location ~ ^/index\.php(/|$) {  
        fastcgi_pass unix:/run/php/php8.1-fpm.sock;  
  
        fastcgi_split_path_info ^(.+\.php)(/.*)$;  
  
        include fastcgi_params;  
  
        fastcgi_param SCRIPT_FILENAME $realpath_root$fastcgi_script_name;  
  
        fastcgi_param DOCUMENT_ROOT $realpath_root;  
  
        internal;  
    }  
  
    location ~ \.php$ {  
        return 404;  
    }  
}
```

Configuration NGINX - Symfony

PM2

Outil qui permet d'avoir ses apps
Node.JS en prod

<https://pm2.keymetrics.io/>



Installer NPM et PM2

Télécharger Node

```
curl -fsSL https://deb.nodesource.com/setup_16.x |  
sudo -E bash -
```

Installer Node

```
apt install nodejs
```

Mettre à jour npm

```
npm install -g npm@9.7.1
```

Installer PM2

```
npm install pm2 -g
```

PM2

Lancer une app

```
pm2 start "npm run start" --name "app_name"  
-u www-data --hp /home/project_dir
```

Lister les apps

```
pm2 list
```

Afficher une app

```
pm2 show [app_name]
```

Supprimer une app de pm2

```
pm2 delete [app_name]
```