|  |  |
| --- | --- |
|  | Partage de fichiers WebDAV |

# WebDAV sur nginx

## Présentation

WebDAV (*Web-based Distributed Authoring and Versioning*) est une **extension du protocole HTTP**.

WebDAV permet de **simplifier la gestion de fichiers avec des serveurs distants**.   
Il permet de **récupérer**, **déposer**, **synchroniser et publier des fichiers** (et dossiers) rapidement et facilement.

L'objectif principal de WebDAV est de **rendre possible l'écriture à travers le web** et pas **seulement la lecture de** **données**. WebDAV permet à plusieurs utilisateurs d'éditer le contenu d'un dossier web simultanément.

Il saura gérer les droits d'accès aux fichiers (ou dossiers), en **verrouillant momentanément les fichiers et dossiers édités**.

Par exemple …

* Il est possible de **partager des documents texte** **ou des chiffriers** avec ses collègues en les téléchargeant sur le serveur WebDAV ;
* Un utilisateur peut aussi être capable de partager sa collection de musique avec ses proches en leur donnant simplement une URL. Tout cela peut être réalisé sans qu'ils n'installent quoi que ce soit.

Il existe plusieurs façons de gérer les fichiers sur un serveur distant.   
WebDAV présente plusieurs avantages par rapport à d'autres solutions telles que FTP ou Samba.

### Utilisation de WebDAV

WebDAV offre plusieurs avantages ...

* **Intégration native sur tous les principaux systèmes d'exploitation** (Windows, Mac, Linux) :   
  Il n'est pas nécessaire d'installer un logiciel tiers pour utiliser WebDAV ;
* **Prise en charge des transferts partiels** ;
* **Plus de choix pour l'authentification** :   
  Être sur HTTP signifie que **NTLM**, **Kerberos**, **LDAP**, ... sont tous possibles.

Selon sa situation, WebDAV peut être la meilleure solution pour ses besoins.

### Étape 01 – Installation de nginx

Pour installer le serveur web ...  
**>>** **sudo** **apt -y update**   
**>>** **sudo apt -y update**  
**>>** **sudo apt -y install nginx-full nginx-extras**

Le serveur Web nginx doit être installé et en cours d'exécution.

**Création des répertoires**

Il faut par la suite désigner un répertoire pour WebDAV.   
Il faudra créer le nouveau répertoire /var/www/webdav pour cela.   
Il faudra également modifier le propriétaire de www-data (l'utilisateur Apache) afin de permettre à Apache d'y écrire.  
**>>** **sudo** **mkdir --verbose /var/www/webdav**  
**>>** **sudo** **mkdir --verbose /var/www/webdav/temp**  
**>>** **sudo chown --verbose --recursive www-data:www-data /var/www/**  
**>>** **sudo chmod --verbose --recursive 755 /var/www/**

**Ajouts de quelques fichiers**

On peut peupler le répertoire /var/www/webdav …

On ajoute quelques fichiers qu’un client peut télécharger …  
**>>** **sudo** **echo "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Proin enim est, blandit in enim eu, ultrices congue velit." | sudo tee /var/www/webdav/lorem\_ipsum\_1**  
**>>** **sudo** **echo "Ut sagittis placerat turpis, pellentesque tempus eros dictum ut. Pellentesque rutrum tincidunt iaculis. Cras elementum posuere eros eu cursus." | sudo tee /var/www/webdav/lorem\_ipsum\_2**  
**>>** **sudo** **echo "** **Vestibulum ac mattis ligula. Donec iaculis orci ac est congue, vehicula ullamcorper metus ultricies. Sed rhoncus malesuada augue nec accumsan." | sudo tee /var/www/webdav/lorem\_ipsum\_3**

### Étape 02 – Authentification de base

Dans un premier temps il faut **générer le fichier** (appelé **users.password**) qui **emmagasine les mots de passe pour les utilisateurs**.   
Dans l'authentification Digest, il existe le **champ domaine qui agit comme un espace de noms pour les utilisateurs**.

Pour cet exemple, le premier utilisateur s'appellera tux.

Pour générer le fichier digest, il faut **installer les dépendances pour nginx** …  
**>>** **sudo apt install apache2-utils  
>>** **sudo** **mkdir --verbose /etc/nginx/webdav**

# Création des utilisateurs

# On doit, dans un premier temps, créer un fichier d'utilisateurs. Ce fichier doit se trouver dans un répertoire lisible par nginx (par exemple le répertoire dans lequel se trouve le site – le répertoire qui est servi par nginx) … >> **sudo** htpasswd -c <Nom de la base de données> <Nom de l’utilisateur> >> **sudo htpasswd -c /etc/nginx/webdav/users.password tux**

L'utilitaire va demander le mot de passe pour l’utilisateur.

**Remarques** …  
Si jamais on souhaite ajouter un utilisateur, on utilisera la même commande sans l'option -c …  
Pour en supprimer un ce sera l'option -D qui sera à utiliser.

Pour afficher la liste des utilisateurs créés …  
**>>** sudo cat /etc/nginx/webdav/users.password

## Mise en place d’un site web sécurisé

Les **fichiers qui seront livré par le serveur** se situent dans le répertoire **sous /var/www/html**.

Par défaut, **nginx s'attend à ce que les fichiers d’un site web se retrouvent dans un répertoire spécifique** (répertoire qui peut varier).   
On détermine cet emplacement dans le fichier de configuration du serveur.

### Étape 03 – Création du certificat et des clés

Le répertoire /etc/ssl/certs, qui peut être utilisé pour contenir le certificat public.   
Ce dernier doit déjà exister sur le serveur.

On crée le répertoire /etc/ssl/private, répertoire qui va contenir le fichier de clé privée.   
Étant donné que le secret de cette clé est essentiel pour la sécurité, de modifier les autorisations pour empêcher tout accès non autorisé …  
**>> sudo mkdir --verbose /etc/nginx/certs  
mkdir: création du répertoire '/etc/nginx/certs'**

La prochaine est la création d’une paire de clé et d’un certificat auto-signé avec OpenSSL en une seule commande …  
**>> sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/nginx/certs/nginx.key \  
 -out /etc/nginx/certs/nginx.crt**

openssl propose une série de questions afin de personnaliser le certificat.

On répond à ces questions de manière appropriée.   
La ligne la plus importante est celle qui demande le nom commun (par exemple, le FQDN du serveur ou son nom).

**Country Name (2 letter code) [AU]:CA  
State or Province Name (full name) [Some-State]:Quebec  
Locality Name (eg, city) []:Montreal  
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Cie Tux Info  
Organizational Unit Name (eg, section) []:Web  
Common Name (e.g. server FQDN or YOUR name) []:www.tux.info  
Email Address []:webmaster@tux.info**

Les deux fichiers qui ont été créés seront placés dans le répertoire /etc/ngginx/certs.

**Remarque** …  
Pour vérifier le contenu du certificat généré …  
**>> sudo** **openssl x509 -noout -text -in /etc/nginx/certs/nginx.crt**

**Secret de transmission parfait**

Lors de l’utilisation d’OpenSSL, il est intéressant de créer également un groupe Diffie-Hellman fort.

Ce groupe est utilisé pour négocier le secret de transmission parfait (*Perfect Forward Secrecy*) avec les clients.  
Le secret de transmission parfait signifie qu'un élément d'un système de chiffrement modifie automatiquement et fréquemment les clés qu'il utilise afin de chiffrer et déchiffrer les informations. Ainsi si la dernière clé est compromise, elle n'expose qu'une petite partie des données sensibles de l'utilisateur.

On peut créer ce secret de transmission parfait …  
**>> sudo openssl dhparam -out /etc/nginx/certs/dhparam.pem 2048**

L’exécution de cette commande peut prendre quelques minutes, mais une fois terminé, on aura un groupe DH puissant dans /etc/nginx/certs/dhparam.pem qu’il sera possible d’utiliser dans la configuration du site Web …  
**>> ls -l /etc/nginx/certs  
total 12  
-rw-r--r-- 1 root root 424 13 nov 09:18 dhparam.pem  
-rw------- 1 root root 1704 12 nov 16:06 monsite.info.key  
-rw-r--r-- 1 root root 1566 12 nov 16:08 monsite.info.pem**

On modifie les permissions sur le répertoires contant les certificats …  
**>> sudo chmod --verbose 700 /etc/nginx/certs  
Le mode de '/etc/nginx/certs' a été modifié de 0755 (rwxr-xr-x) en 0700 (rwx------)**

### Étape 04 – Configuration de WebDAV

**Modification du fichier de configuration de site**

Dans un premier temps, il faut modifier le fichier de configuration dans /etc/apache2/sites-available/000-default.conf en utilisant l'éditeur de texte nano …  
**>>** **sudo mv --verbose /etc/nginx/sites-available/default /etc/nginx/sites-available/default.original  
>>** **sudo nano /etc/nginx/sites-available/default  
server {  
 listen 80 default\_server;  
 listen [::]:80 default\_server;  
 server\_name \_;  
 return 301 https://$host$request\_uri;**

**}**

**server {  
 listen 443 ssl http2;  
 listen [::]:443 ssl http2;  
 server\_name \_;  
  
 root /var/www/webdav/;  
 index index.html index.htm;**

**ssl\_certificate /etc/nginx/certs/nginx.crt;**

**ssl\_certificate\_key /etc/nginx/certs/nginx.key;**

**ssl\_dhparam /etc/nginx/certs/dhparam.pem;**

**ssl\_session\_cache shared:SSL:10m;**

**ssl\_session\_timeout 10m;**

**ssl\_protocols TLSv1 TLSv1.1 TLSv1.2;**

**ssl\_ciphers EECDH+CHACHA20:EECDH+AES128:RSA+AES128:EECDH+AES256:RSA+AES256:EECDH+3DES:RSA+3DES:!MD5;**

**ssl\_prefer\_server\_ciphers on;**

**add\_header Strict-Transport-Security "max-age=31536000";**

**add\_header X-Content-Type-Options nosniff;**

**# access\_log /var/log/nginx/webdav/access.log combined gzip;  
# error\_log /var/log/nginx/webdav/error.log warn;**

**location / {**

**auth\_basic "Restricted site.";**

**auth\_basic\_user\_file /etc/nginx/webdav/users.password;**

**client\_body\_temp\_path /var/www/webdav/temp;**

**dav\_methods PUT DELETE MKCOL COPY MOVE;**

**dav\_ext\_methods PROPFIND OPTIONS;**

**create\_full\_put\_path on;**

**dav\_access user:rw group:rw all:rw;**

**autoindex on;**

**}**

**}**

On vérifie la syntaxe du fichier …  
**>>** **sudo nginx -t  
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok  
nginx: configuration file /etc/nginx/nginx.conf test is successful**

On recharche la nouvelle configuration de nginx …  
**>>** **sudo systemctl reload-or-restart nginx.service**

On vérifie si le service est actif …  
**>>** **sudo systemctl status nginx.service**

Pour s’assurer que tout fonctionne correctement, on peut rapidement tester le serveur webdav à partir d’un fureteur (*browser*).

Une image contenant table

Description générée automatiquement

**Restriction des accès**

Pour rendre le serveur en lecture seule, on doit supprimer, dans le fichier /etc/nginx/sites-available/default, la ligne qui contient **dav\_methods PUT DELETE MKCOL COPY MOVE**;.