

SAÉ S2.03 : Installation de services réseau Solution Complète

Errebache Ahmed & Charpentier Lucas

Introduction

Ce document constitue la deuxième partie du travail réalisé dans le cadre de la SAÉ S2.03 : Installation de services réseaux. Il représente la suite du travail préparatoire dont le but était de comprendre le fonctionnement du serveur web Apache2 sur un système Ubuntu, ainsi que de chercher des solutions théoriques, sans rentrer dans les détails, à différentes requêtes données en énoncé.

L'objectif de cette seconde partie est de constituer un tutoriel concret répondant à ces 7 requêtes. Chaque section présente l'objectif visé, la procédure à suivre, les vérifications à effectuer, ainsi que les éventuelles erreurs à éviter.

Résolution des 7 requêtes

1. Basculer la page de défaut de l'hôte vers /home/dev/www pour que, à une requête du navigateur à l'IP du serveur (127.0.0.1) la réponse soit le code html de la page située à cette nouvelle racine.

Procédure :

- Ouvrir le terminal.
- Entrer la commande suivante afin d'ouvrir le fichier de configuration du site par défaut :

« sudo nano /etc/apache2/sites-available/000-default.conf ».

- Chercher la ligne « DocumentRoot /var/www/html » et la remplacer par « DocumentRoot /home/dev/www ». Enregistrer le document.

- Il faut maintenant déclarer le nouveau dossier comme autorisé dans la configuration Apache. Pour ce faire, éditer le fichier apache2.conf à l'aide de la commande suivante : « sudo nano /etc/apache2/apache2.conf ».

- Vers la fin du document, il se trouve une suite de blocs semblables à celui-ci :

<Directory /home/dev/www>

Options Indexes FollowSymLinks

AllowOverride None

Require all granted

</Directory>

- Il faut y ajouter le bloc :
- <Directory /home/dev/www>
 - Options Indexes FollowSymLinks
 - AllowOverride None
 - Require all granted
- </Directory>

Ce bloc donne à Apache les droits d'accès spécifiques à ce chemin. Enregistrer le document.

- Afin d'appliquer les changements, il est nécessaire de redémarrer Apache à l'aide de la commande suivante : « sudo systemctl restart apache2 ».

Vérification :

Dans le navigateur, se rendre à l'adresse <u>http://127.0.0.1</u>.

La page qui était initialement semblable à celle-ci :

$\leftarrow \rightarrow \mathbf{C}$	○ □ 127.0.0.1 ☆		\boxtimes	٢	മ ≡	
Files	Apache2 Default Page					
	This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubunt systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you car read this page, it means that the Apache HTTP server installed at this site is working properly. You should replace thi file (located at /var/www/html/index.html) before continuing to operate your HTTP server. If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.	tu 1 S				
	Configuration Overview					
	Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is fully documented in /usr/share/doc/ apache2/README.Debian.gz . Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the manual if the apache2-doc package was installed on this server.					
	The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:					
	<pre>/etc/apache2/ apache2.conf ` ports.conf mods-enabled [*.load ` *.conf conf-enabled ` *.conf sites-enabled ` *.conf</pre>					

Affiche désormais le fichier HTML placé dans le dossier /home/dev/www. Dans notre cas :

Ō	127.0.0.1/	× +		~	9	e x
÷	\rightarrow C	Image: 127.0.0.1 5	× ☆	ً	٢	മ ≡
Bo	onjour dep	ouis /home/dev/www				

Il est nécessaire qu'Apache ait les droits de lecture sur le dossier /home/dev/www. Dans le cas inverse, une erreur « Forbidden » apparaitra. Il faut donc entrer les lignes suivantes si ce n'était pas déjà fait :

« sudo chown -R www-data:www-data /home/dev/www »

« sudo chmod -R 755 /home/dev/www »

Il ne faut également pas oublier de redémarrer Apache après toute modification de configuration, ce sera aussi le cas pour les requêtes suivantes.

2. Revenir à la configuration initiale. Ajouter à la racine /var/www/ un lien symbolique vers /home/dev/www baptisé monsite.fr et obtenir en réponse une page de lien des dossiers de la racine, à savoir : icon rep html/ icon rep monsite.fr/ Quand on clic sur monsite.fr/ on a la page attendue.

Procédure :

- Rééditer le fichier 000-default.conf avec la commande : « sudo nano /etc/apache2/sites-available/000-default.conf ».

- Revenir à la ligne « DocumentRoot /home/dev/www » et la remplacer par « DocumentRoot /var/www/ ». Enregistrer le document.

- Dans le terminal, créer un lien symbolique nommé monsite.fr dans /var/www/ : « sudo ln - s /home/dev/www /var/www/monsite.fr ».

- S'assurer qu'Apache peut lister les dossiers dans /var/www/. Pour ce faire, vérifier que le fichier /etc/apache2/apache2.conf contient bien le bloc :

<Directory /var/www/>

Options Indexes FollowSymLinks

AllowOverride None

Require all granted

</Directory>

Et en particulier le terme « Indexes » après « Options ». (« sudo nano /etc/apache2/apache2.conf »).

- Redémarrer Apache pour que les modifications prennent effet. (« sudo systemctl restart apache2 »).

Vérification :

Dans le navigateur, se rendre à l'adresse <u>http://127.0.0.1</u>. La page devrait être semblable à celle-ci :

$\leftarrow \rightarrow C$	0 🗅 127.0.0.1	☆	ල 🕄 දු =
Index o	f /		
Name	Last modified Size Description		
<u>html/</u> 2 <u>monsite.fr/</u> 2	2025-05-16 15:43 - 2025-06-05 20:01 -		
Apache/2.4.58 (U	Jbuntu) Server at 127.0.0.1 Port 80		

En cliquant sur « monsite.fr/ », on accède à la page HTML que nous avions placé dans /home/dev/www.

Si la page par défaut d'Apache s'affiche à la place de l'index, c'est probablement que le terme « /html » a été ajouté à la ligne « DocumentRoot /var/www » comme c'était le cas avant la première modification du fichier 000-default.conf.

Si cliquer sur monsite.fr/ renvoie une erreur « Forbidden », il faut vérifier que le lien symbolique pointe vers un dossier existant et accessible, et qu'il ait les permissions nécessaires, activables avec les commandes : « sudo chown -R www-data:www-data /home/dev/www » et « sudo chmod -R 755 /home/dev/www ».

3. Modifier le fichier /etc/hosts pour associer temporairement un nom de domaine, à l'ip du serveur, à savoir monsite.fr à 127.0.0.1. L'url monsite.fr dans le navigateur donne alors accès au serveur (aux liens des dossiers de la racine).

Procédure :

- Éditer le fichier /etc/hosts à l'aide de la commande suivante dans le terminal : « sudo nano /etc/hosts ».

- Ajouter la ligne « 127.0.0.1 monsite.fr » à la fin du fichier afin d'associer localement le nom de domaine « monsite.fr » à l'adresse IP de loopback donc 127.0.0.1. Enregistrer le fichier.

- Il n'est ici pas nécessaire de redémarrer Apache, cette modification prend effet instantanément.

Vérification :

Dans le navigateur, ouvrir un nouvel onglet et taper <u>http://monsite.fr</u> dans la barre d'adresse. Cette page apparait :

$\leftarrow \rightarrow \mathbf{C}$ \bigcirc \textcircled{B} monsite.fr	<u>ት</u>	ල 💽 එ =
Index of /		
<u>Name Last modified</u> <u>Size</u> <u>Description</u>		
<u>html/</u> 2025-05-16 15:43 - <u>monsite.fr/</u> 2025-06-05 20:01 -		
Apache/2.4.58 (Ubuntu) Server at monsite.fr Port 80		

Si monsite.fr redirige vers un autre site internet, vérifier que l'URL entrée est bien <u>http://monsite.fr</u> et non <u>https://monsite.fr</u>. S'assurer également qu'il n'y a pas de logiciel comme un antivirus qui intercepte les requêtes http.

Si le site n'est tout de même pas accessible, vérifier que la ligne « 127.0.0.1 monsite.fr » a été correctement écrite dans le fichier /etc/hosts.

4. Créez un hôte virtuel afin que le nom de domaine monsite.fr donne accès directement à la page du lien monsite.fr.

Procédure :

- Dans le terminal, créer un nouveau fichier de configuration pour le site monsite.fr en copiant celui par défaut : « sudo cp /etc/apache2/sites-available/000-default.conf /etc/apache2/sites-available/monsite.fr.conf ».

- Modifier ce fichier en l'ouvrant avec la commande : « sudo nano /etc/apache2/sites-available/monsite.fr.conf ».

- Rechercher les 2 lignes commençant par « ServerAdmin » et « DocumentRoot ». Ajouter audessus ou en dessous la ligne « ServerName monsite.fr ». Modifier la ligne « DocumentRoot /var/www/ » par « DocumentRoot /var/www/monsite.fr ». Il faut que DocumentRoot pointe vers le lien symbolique créé précédemment et non vers /home/dev/www. Enregistrer le document.

- Pour activer le nouveau site, utiliser la commande « sudo a2ensite monsite.fr.conf ».
- Redémarrer Apache pour appliquer ces changements. (« sudo systemctl restart apache2 »)

Vérification :

L'adresse <u>http://monsite.fr</u> qui affichait l'index affiche désormais la page html créée au départ dans /home/dev/www via le lien /var/www/monsite.fr.

$\leftarrow \rightarrow C$	○ & monsite.fr	ネ ☆	${igarsing}$	٢	பி	≡		
Boniour depuis /home/dev/www								
Donjour uch								

Erreurs potentielles :

Si le site affiche toujours l'index de /var/www, l'hôte n'a pas été pris en compte. Vérifier que le fichier etc/apache2/sites-available/monsite.fr.conf est bien actif avec la commande « sudo a2query -s monsite.fr.conf ».

5. Modifiez cet hôte virtuel de façon à ce que tous les sous domaines de monsite.fr donnent un accès également.

Procédure :

- Éditer le fichier de configuration du site monsite.fr avec la commande « sudo nano /etc/apache2/sites-available/monsite.fr.conf ».

- Ajouter la ligne « ServerAlias *.monsite.fr » juste en dessous de la ligne « ServerName monsite.fr » créée précédemment. Cela associe tous les sous-domaines à cet hôte virtuel (par exemple : test.monsite.fr ou www.monsite.fr).

Après toutes ces modifications, le fichier devrait ressembler à ceci :

<VirtualHost *:80> # the server uses to identify itself. This is used when creating # value is not decisive as it is used as a last resort host regardless. #ServerName www.example.com ServerName monsite.fr ServerAlias *.monsite.fr ServerAdmin webmaster@localhost DocumentRoot /var/www/monsite.fr # error, crit, alert, emerg. # It is also possible to configure the loglevel for particular ErrorLog \${APACHE LOG DIR}/error.log CustomLog \${APACHE LOG DIR}/access.log combined # include a line for only one particular virtual host. For example the # after it has been globally disabled with "a2disconf". #Include conf-available/serve-cgi-bin.conf /VirtualHost>

- Enregistrer le fichier et redémarrer Apache. (« sudo systemctl restart apache2 »)

- Ajouter les sous domaines souhaités dans le fichier /etc/hosts. Dans notre cas, nous allons créer les sous domaines <u>www.monsite.fr</u> et test.monsite.fr. Pour ce faire, éditer le fichier avec la commande « sudo nano /etc/hosts » puis ajouter les lignes « 127.0.0.1 <nom du sous domaine>.monsite.fr » en dessous de « 127.0.0.1 monsite.fr ». Dans notre cas, l'apparence du fichier est la suivante :



- Enregistrer le fichier. Il n'est pas nécessaire de redémarrer Apache après cette étape.

Vérification :

Entrer dans la barre de recherche les sous domaines créés. Cela mène vers la page de monsite.fr. Dans notre cas les 2 sous domaines affichent ceci :

← → C	O & test-monsite.fr	% ☆	© @ £) =	e → C	⊘ A www.monste.tr	⊼ ☆	© @ £ =
Boniour denuis /home/dev/www			Boniour depuis /home/dev/www				
Donjour				2011jour av	F //		

Erreurs potentielles :

Il est important de déclarer dans le fichier /etc/hosts les sous domaines que l'on souhaite tester.

Si le sous domaine est <u>www.monsite.fr</u>, il est possible que l'entrer dans la barre de recherche renvoie vers le site des pages jaunes. Cela est dû au fait que l'URL force le https au lieu du http. Lancer la recherche dans un onglet de navigation privée règle le problème.

6. Pour le site monsite.fr, on veut que le sous domaine www.monsite.fr dans l'url soit récrit en monsite.fr.

Procédure :

- Dans le terminal, activer le module mod_rewrite avec la commande « sudo a2enmod rewrite ».

- Créer un fichier .htaccess dans le dossier /home/dev/www et l'éditer : « sudo nano /home/dev/www/.htaccess ».

- Y ajouter les 3 lignes suivantes :

- RewriteEngine On
- RewriteCond %{HTTP_HOST} ^www\.monsite\.fr\$ [NC]
- RewriteRule ^(.*)\$ http://monsite.fr/\$1 [L,R=301]

Cela permet de rediriger les requêtes vers <u>www.monsite.fr</u> vers monsite.fr, le tout en conservant le chemin demandé (\$1). Par exemple, www.monsite.fr/accueil redirige vers monsite.fr/accueil.

- Enregistrer et fermer le fichier.

- Éditer le fichier de configuration de monsite.fr (« sudo nano /etc/apache2/sitesavailable/monsite.fr.conf ») en ajoutant le bloc :

<Directory /var/www/monsite.fr>

AllowOverride All

</Directory>

L'instruction AllowOverride All permet à Apache de considérer le fichier .htaccess.

- Enregistrer puis fermer le fichier et redémarrer Apache. (« sudo systemctl restart apache2 »)

Vérification :

Entrer l'URL <u>http://www.monsite.fr</u>. Lorsque la page s'affiche, l'URL devient simplement « monsite.fr » :

C O A monsite.fr X ☆ O Q A ≡

Erreurs potentielles :

Une faute de frappe dans les lignes du fichier .htaccess peut causer une erreur 500 et empêcher la redirection de fonctionner. Vérifier alors la bonne syntaxe du contenu de ce fichier.

Il est également possible que la redirection ne fonctionne pas car le chemin du bloc <Directory> ne correspond pas au DocumentRoot.

7. Enfin, on suppose qu'un service quelconque écoute déjà sur le port 80. On demande de changer le port d'écoute d'Apache et de le passer en 8080.

Procédure :

- Modifier le fichier des ports d'écoute d'Apache avec la commande « sudo nano /etc/apache2/ports.conf ».

- Remplacer la ligne « Listen 80 » par « Listen 8080 » puis enregistrer et fermer le fichier.

- Ouvrir le fichier de configuration de monsite.fr (« sudo nano /etc/apache2/sitesavailable/monsite.fr.conf ») et remplacer la première ligne : « <VirtualHost *:80> » par « VirtualHost *:8080 ». Enregistrer et fermer le fichier.

- Répéter cette opération avec le fichier 000-default.conf (« sudo nano /etc/apache2/sites-available/000-default.conf »).

- Redémarrer Apache. (« sudo systemctl restart apache2 »)

Vérification :

La page à l'adresse 127.0.0.1 qui affichait écouter au port 80, comme le montre l'image ci-dessous :



Affiche désormais écouter au port 8080, comme ci-dessous :



Si la page affiche une connexion impossible, c'est probablement car la recherche est effectuée sous le port 80 par défaut. Il faut donc explicitement rechercher au port 8080 avec « :8080 » à la fin de l'URL.

Il est également possible que le Pare-Feu bloque le port 8080, il faut donc forcer l'autorisation avec la commande « sudo ufw allow 8080 ».

Conclusion

Ce projet nous a non seulement permis de mieux comprendre le fonctionnement et la configuration d'un serveur Apache2, mais nous a également apporté des connaissances dans les services réseaux en général. Les recherches nécessaires pour résoudre ces 7 requêtes ont grandement amélioré notre savoir dans ce domaine, et les compétences acquises comme la manipulation des fichiers de configuration, la gestion des hôtes virtuels, la redirection d'URL ou encore le changement de port du serveur seront réutilisables dans d'autres contextes, en particulier dans le cadre de projets web ou d'hébergement local. On peut également souligner un gain d'aisance dans l'utilisation du système d'exploitation Linux, qui nous sera indispensable dans certains de nos travaux futurs. Cette SAÉ était donc un travail enrichissant dans plusieurs aspects, et pourra servir de tutoriel et de fichier de référence dans le cadre d'une utilisation future d'Apache2.