Table des matières

[1. Intranet.exemple1.lan 1](#_Toc149664869)

[2. port.exemple1.lan sur le port 8080 2](#_Toc149664870)

[3. Site par défaut 5](#_Toc149664871)

[4. page accueil.html 8](#_Toc149664872)

[5. Ecouter des sites dans les dossier personnels des utilisateurs, comme sur un serveur mutualisé 10](#_Toc149664873)

Avez-vous déjà vu …. ? … comment accéder à un site web ?

Précédemment, vous avez créé un serveur Web. Vous avez créé un site internet ‘exemple1.lan’ et vous avez fait en sorte d’y accéder grâce à des noms de site (monsite …) que vous avez paramétré dans le fichier hosts de Windows.

Vous avez donc une idée du fonctionnement d’Apache.

Objectif :

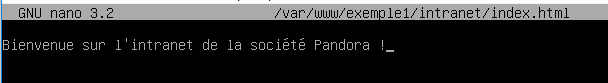
Aller plus loin dans l’utilisation d’Apache. Utiliser les différentes configurations des virtualhost que vous avez entraperçus précédemment.

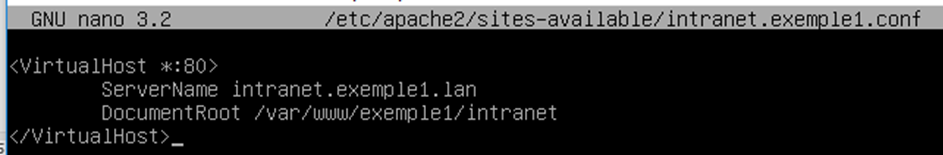
Au travail !!!

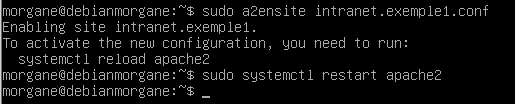
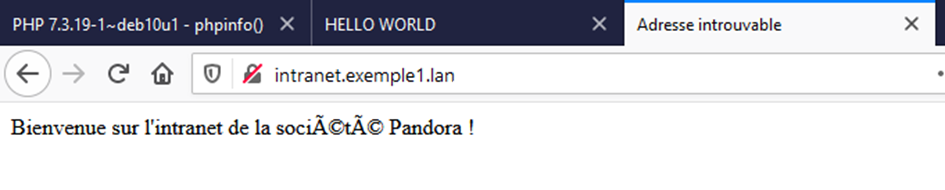
En utilisant les recherches que vous avez faites précédemment, le cours sur Apache, internet, vos connaissances, votre logique, mettez en place les demandes suivantes sur votre serveur :

## Intranet.exemple1.lan

♦Créez un autre site, un site enfant ‘intranet.exemple1.lan’. En affichant la page, vous devez voir écrit : « Bienvenu sur l’Intranet de la société Pandora ! ».

* Par convention, on créer le dossier pour accueillir la page enfant d’un site parent dans le dossier du site parent (exemple1 est le site parent, intranet est le site enfant, l’arborescence devra être /var/www/exemple1/intranet tandis que l’adresse internet qu’on tapera sur le navigateur sera <http://intranet.exemple1.lan>).
* => Commençons par créer le contenu de notre site comme indiqué :
* 

Puis à configurer le virtual host :

* Le port qu’on vous demande : 80, le server name demandé (<http://intranet.exemple1.lan>) et le chemin du dossier qui contient notre site (/var/www/exemple1/intranet).
* On active notre site internet et on redémarre les services Apache pour prendre en compte ce nouveau site internet et sa configuration.
* Il n’y a plus qu’à enregistrer le nom de votre site internet sur le fichier hosts de votre machine Windows (ou votre serveur DNS) et ensuite à tester sur votre navigateur web d’accéder à votre site internet.
* 

## port.exemple1.lan sur le port 8080

♦Créez un autre site, un site ‘port.exemple1.lan’, qui sera hébergé sur le port 8080.

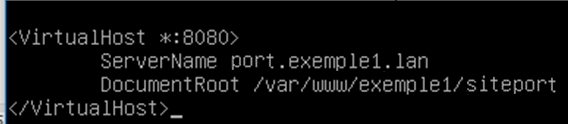
À vous de trouver comment faire cette configuration, comment Apache écoute le nouveau port, et comment spécifié lors de la tentative d’accès depuis un client le port.

En affichant la page, vous devez voir écrit : « Bienvenu sur le site internet sur un port spécifique ! ».

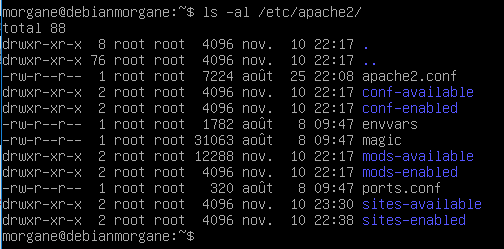
=> Commençons par créer le contenu de notre site comme indiqué :



Écrivons le contenu approprié dans le fichier index.html de siteport.

Puis à configurer le virtual host : 

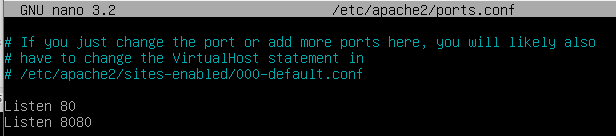
Le port qu’on vous demande : 8080, le server name demandé (<http://port.exemple1.lan>) et le chemin du dossier qui contient notre site (/var/www/exemple1/siteport).

Comme précisé dans les explications, saisir le numéro du port dans le virtual host ne fait pas tout, il faut aussi dire à Apache d’écouter les requêtes qui arrivent sur ce nouveau port.   
Si vous faites un ls -al /etc/apache2, puisqu’au niveau d’apache tout se passe dans ce dossier, vous voyez :

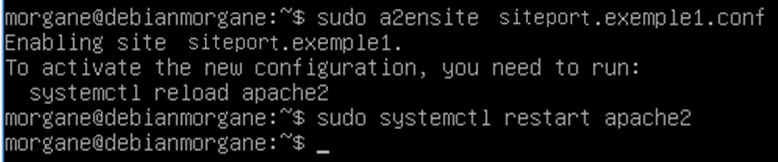
En réfléchissant un peu, quel fichier permet de configurer l’écoute de ce nouveau port ?

Le fichier « ports.conf » bien sûr !



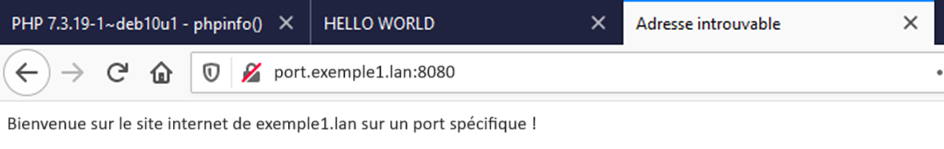
On voit que dans ce fichier, il y a une ligne (Listen 80) qui dit que le serveur apache écoute sur le port 80, et une autre sur le port 443 (Listen 443). Étant donné que nous on veut que le serveur écoute sur le port 8080, qu’est-ce qu’on fait ? On rajoute une ligne « Listen 8080 ». Aussi simple que cela !

On active notre site internet et on redémarre les services Apache pour prendre en compte ce nouveau site internet et sa configuration.



Il n’y a plus qu’à enregistrer le nom de votre site internet sur le fichier hosts de votre machine Windows ou votre serveur dns et ensuite à tester sur votre navigateur web d’accéder à votre site internet. Mais souvenez-vous, il faut préciser à votre navigateur qu’il doit accéder à ce site internet sur le port 8080 !

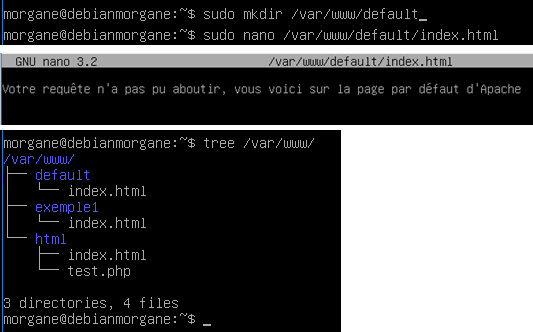
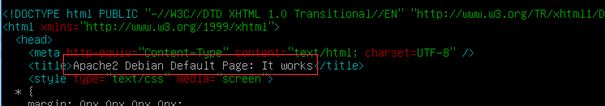
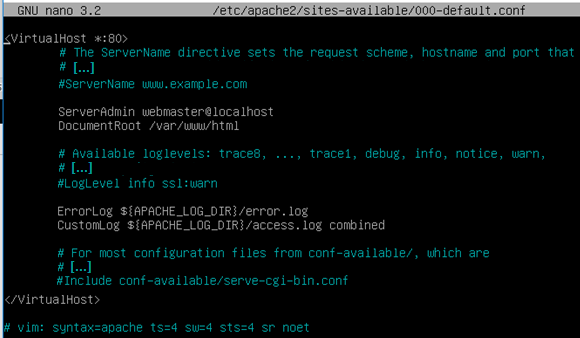
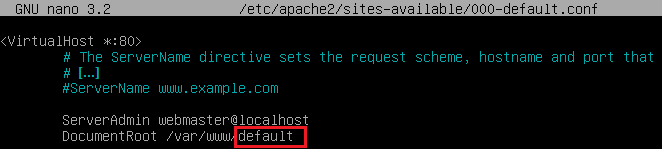
Pour cela, vous devez simplement faire suivre l’adresse web par le port souhaité, en séparant les deux éléments par deux points => http://port.exemple1.lan:8080

Ça fonctionne ! 

Remarquez que si vous ne précisez pas le port sur le chemin HTTP, vous atterrissez où ? Sur la page par défaut ! Et oui, elle est là pour ça, en cas d’erreur elle s’affiche ! 

## Site par défaut

♦Créez un site accessible sur le port 80 nommé ‘default’. Sur sa page, écrivez « Votre requête n’a pas pu aboutir, vous voici sur la page par défaut d’Apache ». Mettez ce site en tant que site par défaut d’Apache.

* Tester que vous accédez à ce site en saisissant dans la barre d’adresse de votre navigateur l’adresse IP de votre. Remarquez que pour réaliser cela, vous n’avez pas besoin d’aller modifier le fichier hosts car il n’y a pas besoin d’associer de nom à l’adresse IP.
* => Créez un site accessible sur le port 80 nommé ‘default’ (sudo mkdir /var/www/default)
* Sur sa page (index.html), écrivez la phrase demandée (sudo nano /var/www/default/index.html et vous savez écrire le contenu)
* 
* => Mettez ce site en tant que site par défaut d’Apache :
* Pour cela, il faut en premier trouver où la vraie page par défaut est paramétrée.
* Regardons le contenu de l’autre site internet présent dans /var/www, il s’agit du site html. En faisant un nano, on voit le contenu du site, on voit dès le début du document que le titre de ce site dans le dossier html est « Apache 2 Debian Default Page : It works », il s’agit donc de la page que l’on voit s’afficher quand on tape l’adresse IP de notre serveur sur un navigateur web. La petite page rose !
* Regardons du côté des virtuals hosts :
* 3 fichiers, 1 que nous avons mis en place, et 2 par défaut.
* Si on regarde le contenu de ce fichier, on reconnait des éléments que l’on connait :   
  <Virtualhost \*80>  
  DocumentRoot /var/www/html  
  </VirtualHost>
* Les autres lignes, vous pouvez chercher leurs significations sur les docs d’apache, pour assouvir votre curiosité. Juste avec celles listées ci-dessus, c’est suffisant. En effet, on voit que ce fichier 000-default qui est donc le fichier de configuration virtual host du site par défaut envoi les requêtes sur les fichiers contenus dans /var/www/html, dossier qu’on a vu juste avant qui contient bien la page par défaut qu’on connait.
* Étant donné que nous avons déjà créé le contenu de notre site, et le fichier de virtual host existe, on l’a sous les yeux, il ne reste plus qu’à le modifier pour correspondre à nos besoins :
* Redémarrez les services apache :
* => Testez que vous accédez à ce site en saisissant dans la barre d’adresse de votre navigateur l’adresse IP de votre serveur.
* Ça fonctionne !

## page accueil.html

♦Par convention on nomme les pages d’accueil index.html ou index.php parce que les serveurs Web savent, grâce une liste, que ce sont les fichiers avec ces noms-là qu’ils doivent ouvrir en premier quand ils accèdent à un site. Créez donc un nouveau site, dont la page d’accueil s’appellera cette fois accueil.html. Évidemment, pour qu’elle soit ouverte par défaut par votre serveur, il faut trouver cette fameuse liste, et rajouter le nom de votre fichier dedans !

Créons le contenu de notre site :





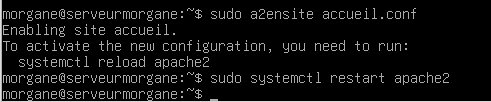


Créons le virtual host du site :

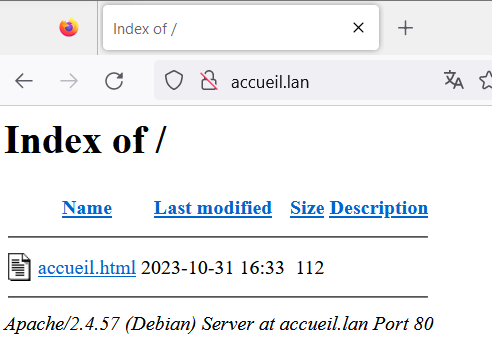




Activons le site :

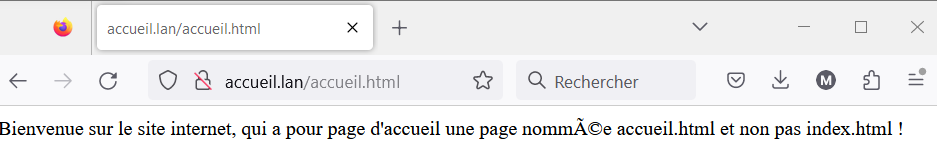


Après avoir enregistré le nom de votre site dans votre serveur dns, testez un accès dessus :



Le site est reconnu, mais s’affiche uniquement le nom de la page, car votre serveur ne sait pas qu’il doit ouvrir par défaut le fichier accueil.html.

Si vous cliquez dessus, ça donne :



Vous accédez bien au contenu de votre site.

Mais notre objectif, c’est que la page accueil.html s’ouvre toute seule.

Pour cela, on sait que sur le serveur Apache, il y a une liste des fichiers que le serveur sait ouvrir par défaut. On sait que cette liste contient au moins le fichier index.html. Il faut donc rechercher sur notre machine linux, dans le dossier apache, pour cibler la recherche, le fichier qui contiendrait ce mot, et donc potentiellement cette fameuse liste, afin qu’on y rajoute notre propre nom.

Pour cela, la commande qui sert à rechercher un fichier par son contenu c’est grep, elle s’utilise de telle façon :

grep [mot à chercher] -r [dossier ou faire la recherche]

donc nous on veut : grep index.html -r /etc/apache2 et ça donne :



Le seul fichier qui ressort de la recherche c’est le fichier /etc/apache2/mods-available/dir.conf avec la variable de configuration DirectoryIndex et la liste qui suit : index.html … index.php …

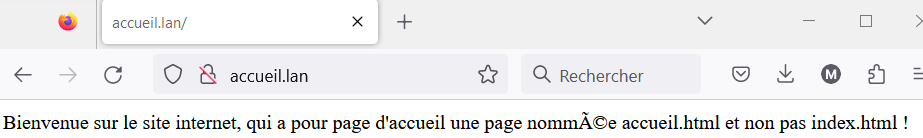
Cela semble donc totalement correspondre à notre besoin. Du coup on va ouvrir le fichier, chercher cette ligne, et rajouter à la fin le nom de notre fichier d’accueil.



Il reste à redémarrer le service Apache pour prendre en compte les modifications, et à tester le bon fonctionnement.



Maintenant, en accédant uniquement à notre site web, la page s’affiche automatiquement, on n’a pas besoin de cliquer dessus.



## Écouter des sites dans les dossiers personnels des utilisateurs, comme sur un serveur mutualisé

♦Configurer votre serveur pour accéder à des sites présents dans /home/[users].  
Ce fonctionnement est celui qu’on retrouve sur les hébergements mutualisés. C’est une configuration à activer dans le fichier de configuration principal d’apache.

On créer d’abord le contenu de notre site internet dans notre dossier personnel sur notre machine linux :



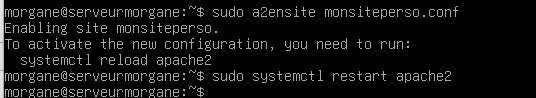


Ensuite, on créer le virtualhost sur apache, pour indiquer que ce site internet existe :



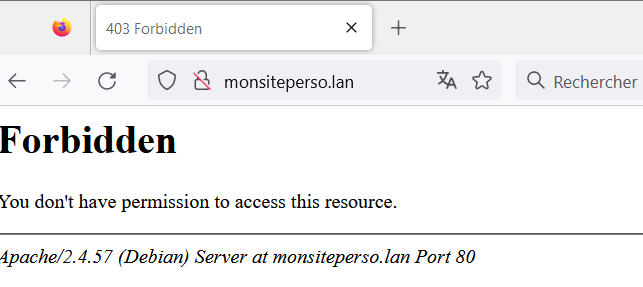


On active le site et redémarre apache :



On enregistre sur votre machine windows dans le fichier hosts ou dans le dns le nom de votre site, et on teste l’accès.

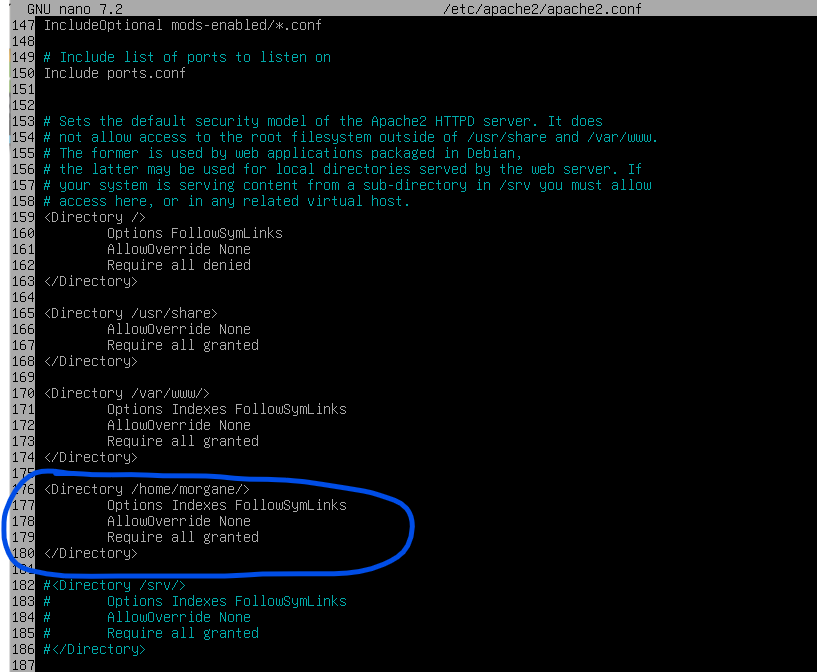
Le site est reconnu, mais vous n’avez pas le droit d’y accéder, car apache n’a pas encore le droit de vous afficher des sites s’ils sont contenus dans /home.



On va donc autoriser cela.   
Ça ce passe dans le fichier apache2.conf : 

De la ligne 170 à 174, à peu près, vous pouvez voir comment le serveur apache autorise les sites s’ils sont configurés dans le dossier /var/www/.

on va donc recopier cette configuration et l’adapter pour autoriser les sites sur notre dossier perso. Ici j’ai juste mis le dossier de mon utilisateur propre, pour éviter que tout le monde puisse faire des sites. Mais si on veut donner la possibilité à tous les utilisateurs, on a juste à changer le chemin en /home tout court.



Lorsque vous avez écrit cette configuration, vous relancez le service apache.

Il reste une dernière modif à faire. Donne le droit aux « autres » d’accéder à votre dossier personnel et donc à votre site, en lecture et exécution, sinon, jamais le serveur web ne pourra afficher le contenu.

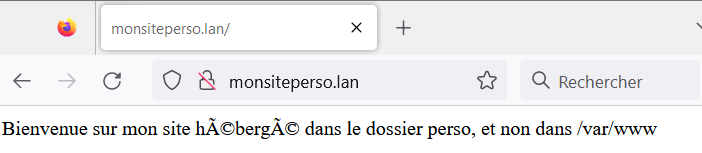
Pour cela, un petit

chmod -R 755 /home/morgane/

permettra de répondre à ce besoin.

Ensuite, depuis votre poste client essayé d’accéder à votre site, ça fonctionne.

Vous avez donc fait un site internet hébergé dans votre dossier personnel, et non pas dans /var/www et vous savez comment faire écouter apache, des sites qui sont dans d’autres dossiers !



♦Créer un autre site web, qui répond au nom par exemple de site2.mondomaine.lan et qui doit contenir 3 pages :

* Accueil
* Contenu
* Contact

Le contenu des pages importe peu, une phrase suffit. Concentrez-vous sur l'organisation des dossiers et de la configuration de votre site.

Pour info, la page accueil sera accessible sous le nom du site donc : site2.mondomaine.lan.  
Et les pages contenu et contact seront logiquement accessibles sous les adresses suivantes : site2.mondomaine.lan/contenu.html et site2.mondomaine.lan/contact.html

♦Complétez votre site 2.

- Mettre une ou plusieurs images sur la page d'accueil

- Créer un dossier "images" pour cela

- L'une de ces images doit rediriger vers la page "Contenu"

- Créer un menu pour accéder aux différentes pages

- Organiser et créer un formulaire pour la page "Contact"

- Créer un dossier "forms"

- Dans le menu, rajouter un bouton qui renvoie vers le site 1

Testez ces fonctionnalités depuis votre client. Vous êtes en train d’organiser votre site, avec des dossiers pour différentes gestions, et des liens entre les pages, vous êtes donc en train de découvrir comment naviguer de page en page et comment s’écrivent ces liens.