Serveur Web Apache et SSH

1. Introduction :

Dans le vaste monde de l'hébergement web et de la gestion de serveurs, deux éléments cruciaux se démarquent pour assurer la sécurité, l'efficacité et la flexibilité : SSH (Secure Shell) et le serveur Apache2. Ces technologies, bien établies dans le domaine de l'informatique, permettent aux administrateurs système et aux développeurs de gérer et de déployer des applications web de manière fiable et sécurisée.

Au cœur de toute infrastructure informatique se trouve la nécessité de gérer à distance les serveurs tout en garantissant la sécurité des échanges d'informations. C'est là que SSH entre en jeu. En fournissant une connexion chiffrée entre un client et un serveur, SSH assure la confidentialité des données, l'authentification sécurisée grâce à l'utilisation de clés cryptographiques, et la possibilité de créer des tunnels sécurisés pour le transfert de données sensible. Que ce soit pour exécuter des commandes à distance, transférer des fichiers de manière sécurisée ou assurer une gestion distante robuste, SSH s'impose comme un pilier de la connectivité sécurisée.

D'un autre côté, pour livrer des contenus web et fournir des applications en ligne, Apache2 se positionne en tant que serveur web de choix. En tant que successeur de la version originale d'Apache, Apache2 excelle dans la gestion des requêtes HTTP, offrant une configuration flexible via des modules extensibles et des fichiers de configuration. La robustesse de sa sécurité, la possibilité de restreindre l'accès aux ressources, et la gestion des certificats SSL/TLS en font un outil essentiel pour la mise en place de sites web dynamiques et sécurisés.

1. SSH et Apache
2. SSH (Secure Shell)
   1. Définition

Par défaut, si on veut communiquer à notre serveur, on ne va pas directement taper sur le serveur. Considerons que celui-ci se situe quelque part loin de chez nous alors qu’on a besoin d’y acceder à distance, pour cela on va utiliser le protocole ssh.

Dans le cas général, ce protocole est déjà installé par les serveurs d’hebergement que l’on utilise, mais dans notre cas, on utilisera une machine virtuel sur virtualBox comme serveur d’hebergement.

* 1. Configuration sous Ubuntu

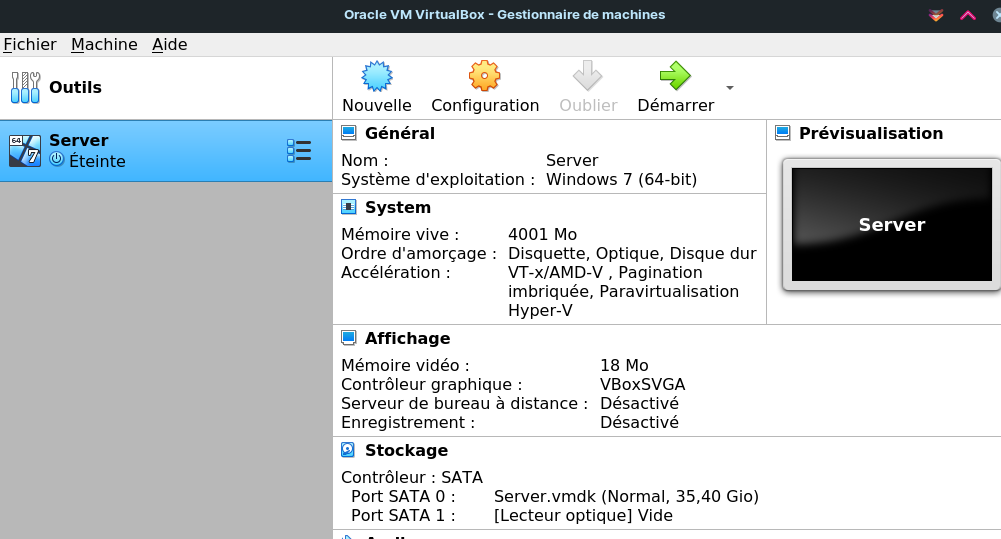
Pour débuter notre configuration, on aura besoin d’installer tout d’abord dans notre machine virtuelle SSH par la commande suivante:

**- sudo apt update** puis **sudo apt install ssh**

Ensuite, on doit configurer le serveur pour que notre machine hôte peut communiquer avec la machine virtuelle car notre machine hôte est par défaut connecter à Internet.

**1ère Etape:** Modifions les paramètres du réseau de la machine virtuelle:

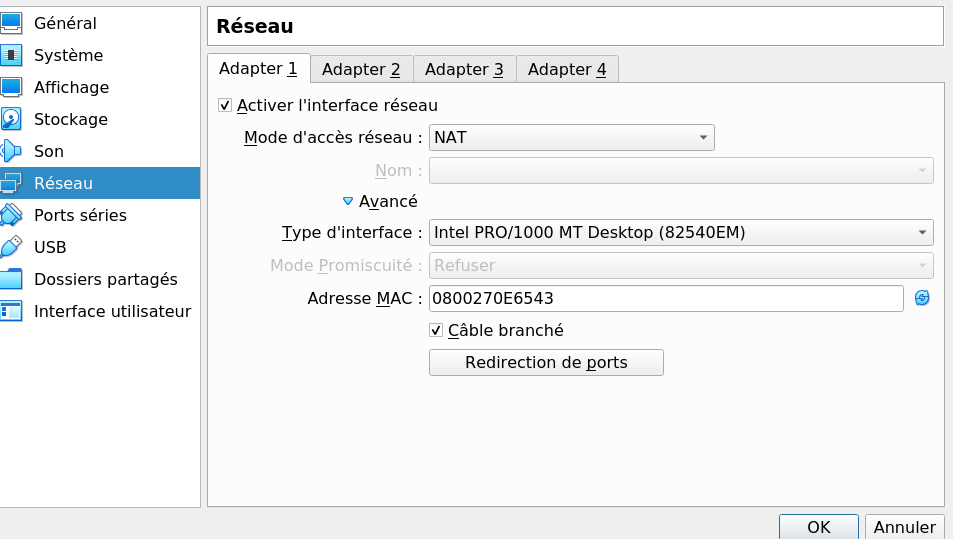
- Allez dans Configuration



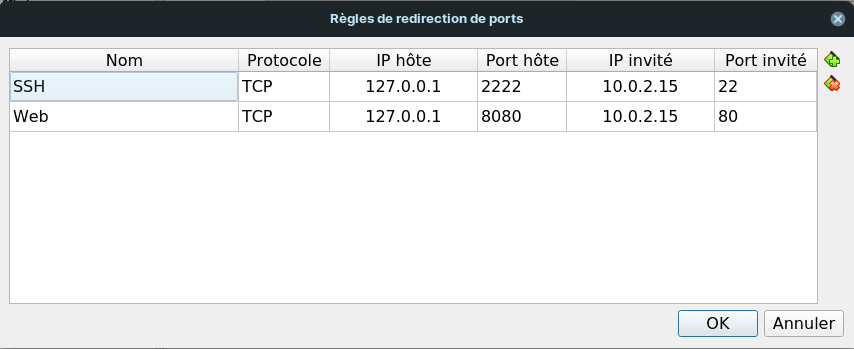
- Puis entrer dans réseau



- Ensuite cliquer sur Avancée



- Cliquer sur redirections de ports



1. Apache2, Installation , lancement d’Apache sous Linux (Ubuntu) .

2.1 Définition et installation

Apache est un logiciel de serveur web open source developpé par Apache Software Foundation. Ce programme est utilisé pou héberger des sites web et des applications web. Il supporte aussi des fonctionnalités telles que la gestion des connexions, la sécurité, et la prise en charge de divers protocoles comme HTTP et HTTPS. On le combine souvent avec des bases de données comme Mysql, ainsi que des langages scripts tels que PHP, Perl ou Python côté serveur. Apache est très populaire et fait partie integrante de nombreuses suites de logiciels destinées au développement web, dont les plus connues sont XAMP, LAMP, MAMP, XAMPP.

Pour installer Apache seul, il suffit d’installer simplement le paquet apache2 avec la commande ‘**sudo apt install apache2**’. Voici quelques commandes utiles pour le lancement d’ Apache :

* Pour démmarer apache2: sudo systemctl start apache2
* Pour arrête apache2: sudo systemctl stop apache2
* Pour relancer apache2: sudo systemctl restart apache2
* Pour recharger la configuration d’apache2: sudo systemctl reload apache2
* Pour tester l’ensemeble de la configuration d’Apache: sudo apache2ctl -t
* Pour tester la configuration des hôtes virtuels: sudo apache2ctl -t -D DUMP\_HOSTS
* Pour tester les hôtes virtuels chargés: sudo apache2ctl -s
* Pour tester les hôtes virtuels chargés et leurs configurations: sudo apache2ctl -S

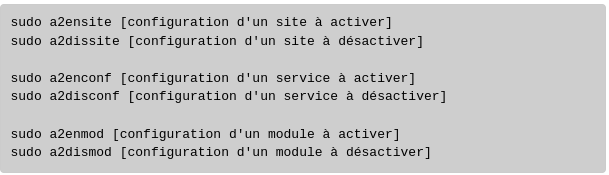
1. Fichiers de configuration:

La configuration d’Apache2 est morcelée c’est-à-dire qu’elle est divisée en plusieurs fichiers et répertoires, dont chacun est responsable d’une partie spécifique de la configuration du serveur web. Cette approche offre plusieurs avantages, notamment la modularité, la facilité de gestion et la possibilité d’apporter des modifications ciblées sans affecter l’ensemblel du serveur.Toutesfois, tous les fichiers de configuration se situent dans répertoire /etc/apache2. Voici quelques éléments clés qui illustrent ce répertoire dans la configuration d’Apache2:

* **sites-available:** qui contient les fichiers de configuration des sites disponible.
* **sites-enable:** qui contient des liens symboliques vers les configurations, dans ‘**sites-available**’, des sites activés.
* **conf-available:** qui contient les fichiers de configuration des autres services disponibles.
* **conf-enabled:** qui contient les liens symboliques vers les configurations, dans ‘**conf-available**’, des autres services activés
* **mods-available:**qui contient les fichiers de configuration des modules d’Apache disponibles
* **mods-enable:**qui contient des liens symboliques vers les configurations, dans ‘**mods-available**’, des modules activés
* **ports.conf:** permet de spécifier les ports à écouter.Il s’agit des ports 80 (port part défaut pour HTTP), et 443 (port par défaut pour HTTPS si le module SSL est activé.

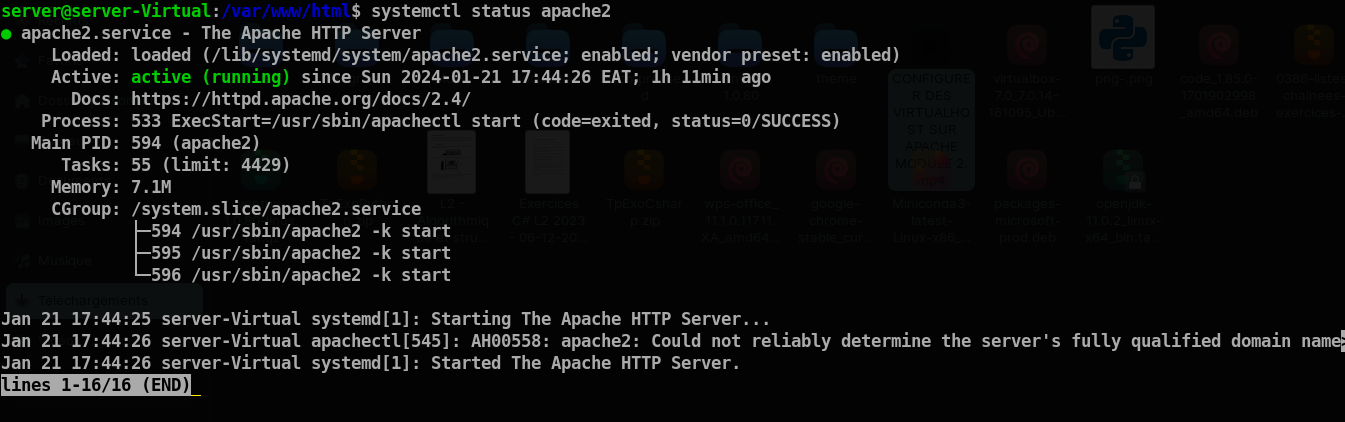
**Remarque**: Normalement les fichiers de configurations globale ‘**apache2.conf**’, ‘**envars**’ n’ont pas à être modifiés. Ces fichiers généraux sont susceptibles d’être écrasés lors d’une mise à jour majeure de son système, et on perd dans ce cas sa configuration personnalisée.

Pour activer ou desactiver ces divers configurations pour Apache2, voici les commandes :

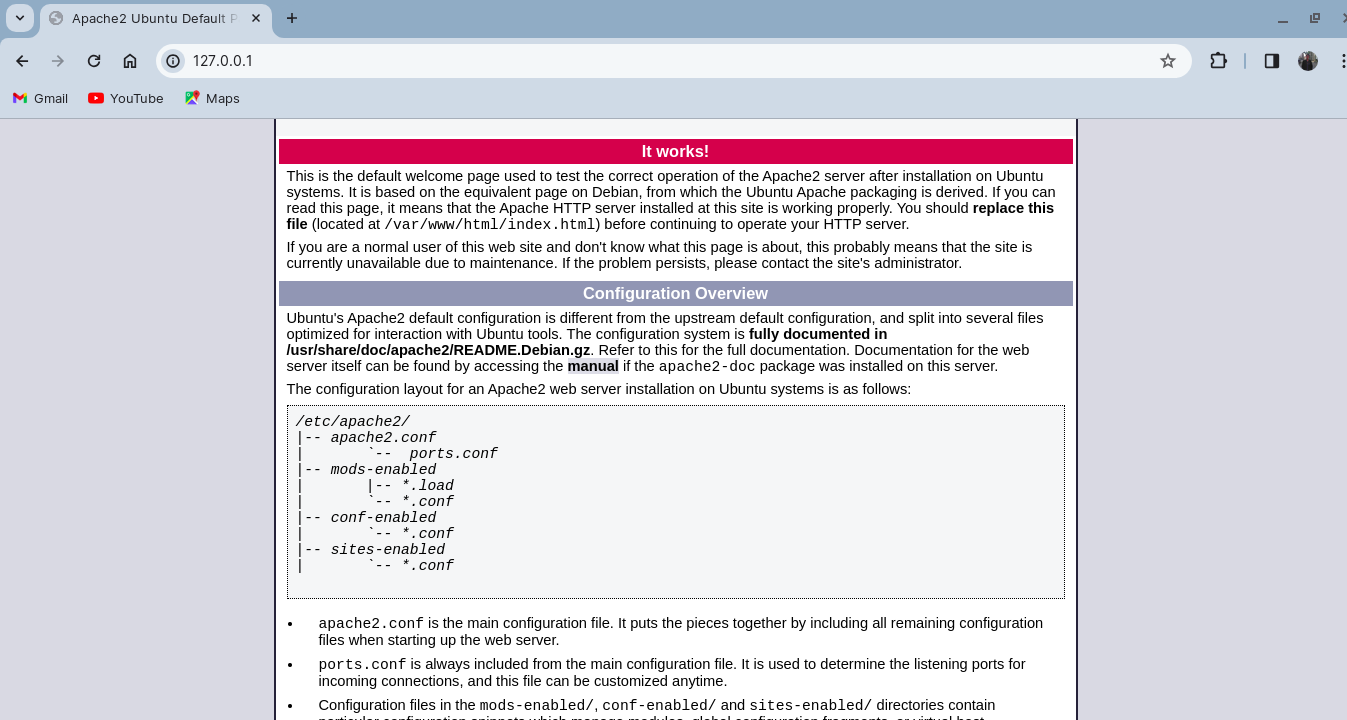


2.2 Démo et Configurations:

Pour commencer, on vérifiera le status d’Apache2 avec la commande suivant:

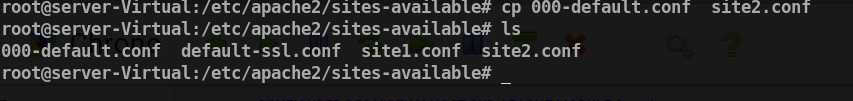


Pour tester si apache2 marche bien et que la machine hôte peut bien acceder au serveur, tapons **l’adresse IP de la machine virtuel 127.0.0.1** dans notre navigateur et cela nous renvoyera la page par défaut d’apache



Puis, comme exemple, on va créer maintenant deux dossiers contenant les fichiers sources des sites à héberger dans notre serveur dans le répertoire **/var/www/html/** avec la commande dans le terminal **mkdir nom\_dossier**, puis plaçons nos fichiers sources dans ce dossier créé.

Maintenant, créer un fichier de configuration dans le répertoire **/etc/apache2/sites-avalaible**



Dans notre cas, on a copié les configurations par défaut d’apache dans notre nouveau fichier **site2.conf.**

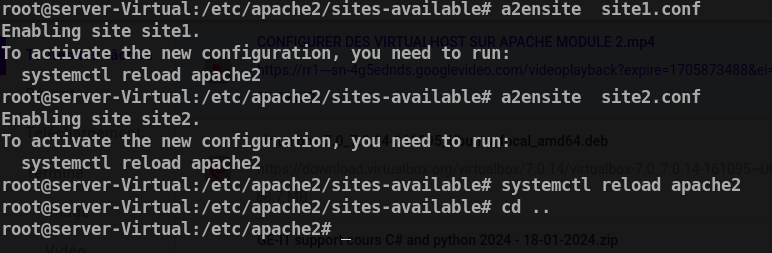
Après avoir créé notre fichier de configuration dans le répertoire /etc/apache2/sites-available, nous pouvons poursuivre en configurant un exemple simple. Voici un exemple de configuration de base pour notre fichier site2.conf :



Dans cette configuration :

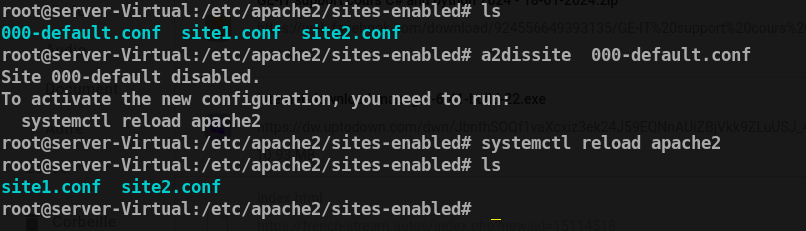
* ServerAdmin : Adresse e-mail de l'administrateur du serveur.
* ServerName : Nom de domaine principal du site.
* ServerAlias : Alias facultatif, permettant de définir des alias supplémentaires pour le site.
* DocumentRoot : Chemin vers le répertoire contenant les fichiers du site.

Une fois que nous avons configuré notre fichier site2.conf, nous devons l'activer avec la commande suivante :



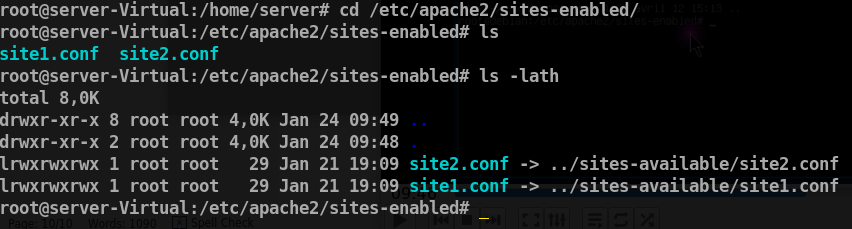
Après cette activation, il est recommandé de reloader Apache2 pour que les mises à jours sont prises en configuration complètes. La commande **systemctl reload apache2** permet de charger Apache.

Puis on peut vérifier les derniers mises à jours de la configuration si on accède le répertoire **cd /etc/apache2/sites-enable** avec **ls.**



A l’inverse, on peut aussi désactiver les sites actifs que nous n’avons pas besoin au cours de la configuration avec la commande **a2dissite nom\_fichier.conf**

**NB: Ce sites-enabled** active seulement des liens symboliques.



Une fois que nous avons terminé de désactiver ou d'activer les sites nécessaires, nous pouvons également vouloir vérifier la syntaxe de nos fichiers de configuration Apache2 pour s'assurer qu'il n'y a pas d'erreurs de syntaxe qui pourraient empêcher Apache2 de démarrer correctement.

Pour ce faire, nous utilisons la commande suivante :

Cette commande vérifie la syntaxe de tous les fichiers de configuration Apache2 et affiche un message indiquant s'il y a des erreurs de syntaxe ou non. Si aucun message d'erreur n'est renvoyé, cela signifie que la syntaxe est correcte et que nous pouvons redémarrer Apache2 en toute sécurité.



Pour redémarrer Apache2 après avoir effectué des modifications de configuration, nous utilisons la commande suivante :



Cela redémarre le service Apache2, appliquant ainsi toutes les modifications de configuration que nous avons apportées.

Enfin, pour garantir que Apache2 démarre automatiquement au démarrage du système, nous pouvons utiliser la commande suivante :



Cela configure le service Apache2 pour qu'il démarre automatiquement lors du démarrage du système.

Une fois toutes ces étapes terminées, notre configuration Apache2 est complète. Nous avons activé et désactivé les sites selon nos besoins, vérifié la syntaxe de nos fichiers de configuration, redémarré Apache2 pour appliquer les modifications et configuré Apache2 pour qu'il démarre automatiquement au démarrage du système. Notre serveur Apache2 est désormais prêt à héberger des sites Web.