DDWS

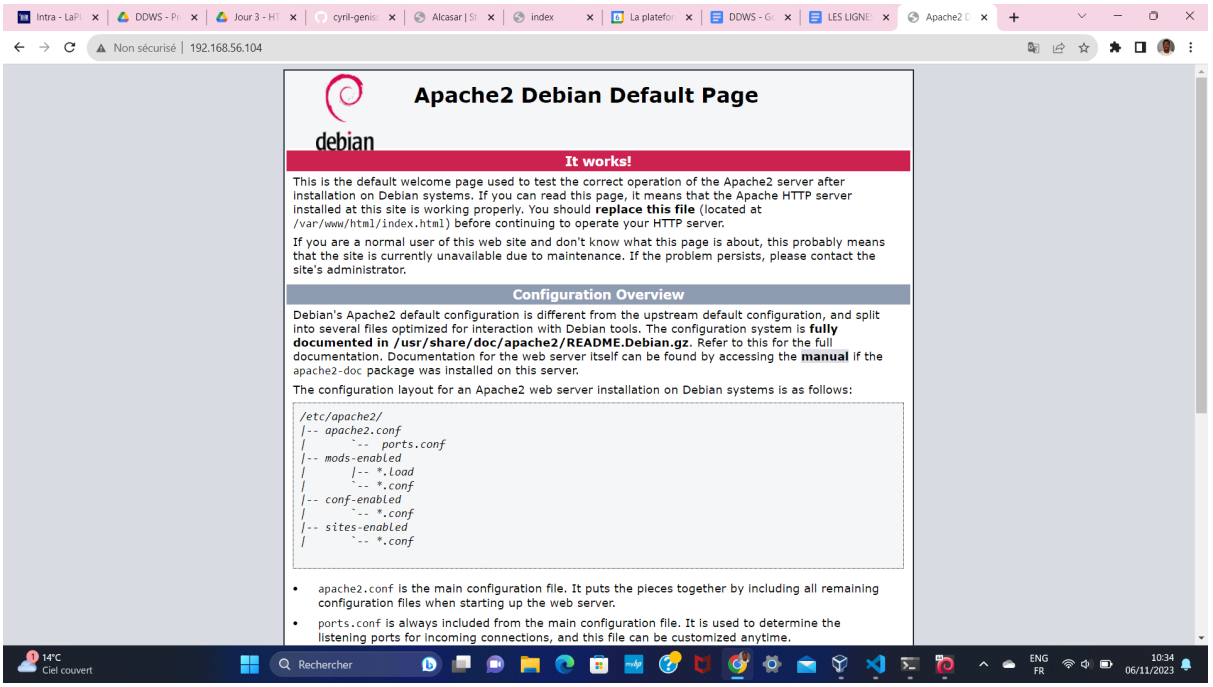
JOB 1

- Installation d'une VM debian avec interface graphique
- Configuration du serveur SSH lors de l'installation

JOB 2

Après l'installation de la VM il faut les étapes suivantes :

- Installer apache2 en exécutant la commande : *sudo apt-get install apache2*
- Pour avoir l'image indiquée dans le sujet qui montre graphiquement que ça marche, il faut ouvrir un navigateur dans la VM ou sur notre machine hôte et taper l'adresse IP de cette dernière



JOB 3

Un serveur web est un logiciel qui permet de transmettre des pages web aux clients qui en font la demande.

Les différents types de serveurs web et leurs avantages & inconvénients :

Serveurs	Avantages	Inconvénients
Apache	Flexible, adaptable à plusieurs systèmes d'exploitation, Large gammes de langage de programmation tels que PHP, Python ou perl ce qui facilite le développement.	Utilisation intensive des ressources, configuration parfois complexes, gestion des connexions concurrentes, redémarrage nécessaires pour appliquer la configuration, réactivité aux changements de trafic, une sécurité moins renforcée, mise à jour moins fréquente.
Nginx	efficacité et rapidité dans le traitement simultané d'un grand nombre de requêtes, capacité de gérer une grande charge élevée tout en restant optimale, il est privilégié par des sites à fort trafic tels que les plateformes e-commerce ou les médias en ligne.	Complexité de la configuration, moins de flexibilité pour l'exécution des scripts, pas de prise en charge native de .htaccess, moins de modules disponibles, besoin de recharger la configuration pour prendre effet.
IIS (Microsoft's Internet Information Service)	Intégration étroite avec les outils Microsoft ainsi qu'avec Active Directory pour une gestion avancée des utilisateurs et des autorisations, sécurité renforcée grâce aux fonctionnalités tels que la prise en charge native du protocole SSL/TLS.	Fonctionne principalement qu' avec les produits Windows Server et . NET Framework.
Lighttpd	Faible consommation en ressources systèmes grâce à son architecture optimisée, hautes performances, efficacité dans la gestion des fichiers statiques.	Moins de fonctionnalités avancées, documentation moins étendue, possibilité de bugs et problèmes de stabilité, complexité pour les utilisateurs débutants.
Caddy Server	Son principal atout réside dans son interface conviviale qui facilite grandement la configuration et l'administration du serveur web, gestion automatique des certificats SSL grâce Let's Encrypt qui favorise considérablement le	Licence de redistribution, limitation de la version gratuite, dépendance à un fournisseur tiers pour le certificat SSL/TLS, moins de modules et d'extensions, communauté moins

	processus de sécurisation des sites webs.	étendue, complexité pour les utilisateurs débutants
Google web server (GWS)	Possibilité d'utiliser des images disque stockées pour créer des instances, Capacités à la demande pour lancer et terminer des instances, Possibilité de marquer vos instances, Variété de systèmes d'exploitation disponibles qui peuvent être installés sur votre instance.	Manque de transparence, personnalisation limitée, dépendance envers google, limitation de la compatibilité, difficulté d'intégration avec d'autres technologies, limitation de support et de documentation.
Litespeed	Haute performance, économie de ressources, compatibilité avec apache, gestion de connexions concurrentes, sécurité améliorée, intégration avec des technologies populaires facilité de configuration.	Coût élevé, licence propriétaire, dépendance vis-à-vis du fournisseur, migration potentielle difficile, limitation de la documentation et des ressources communautaires.
Cherokee	Performance et efficacité, gestion de la charge de travail, configuration graphique conviviale, sécurité renforcée, compatibilité avec les technologies modernes, configuration flexible, communauté active, gestion des certificats SSL/TLS, documentation complète.	Moins de modules et d'extensions, possibilité de bugs et problèmes de stabilité, complexité pour les utilisateurs débutants, mise à jour moins fréquente.

JOB 4

- Lancement du serveur web apache2. Pour le rendre opérationnel, il faut passer par le serveur web bind9. Pour ce faire il faut :

➢ Installer bind9

➢ configurer ensuite bind9.

➢ La configuration *options de bind9 ensuite;

➢ La configuration *locale de bind9;

➢ La création du fichier du fichier zone

- On vérifie après si les configurations ont été bien faites par les commande suivantes :

  named-checkconf /etc/bind/named.conf

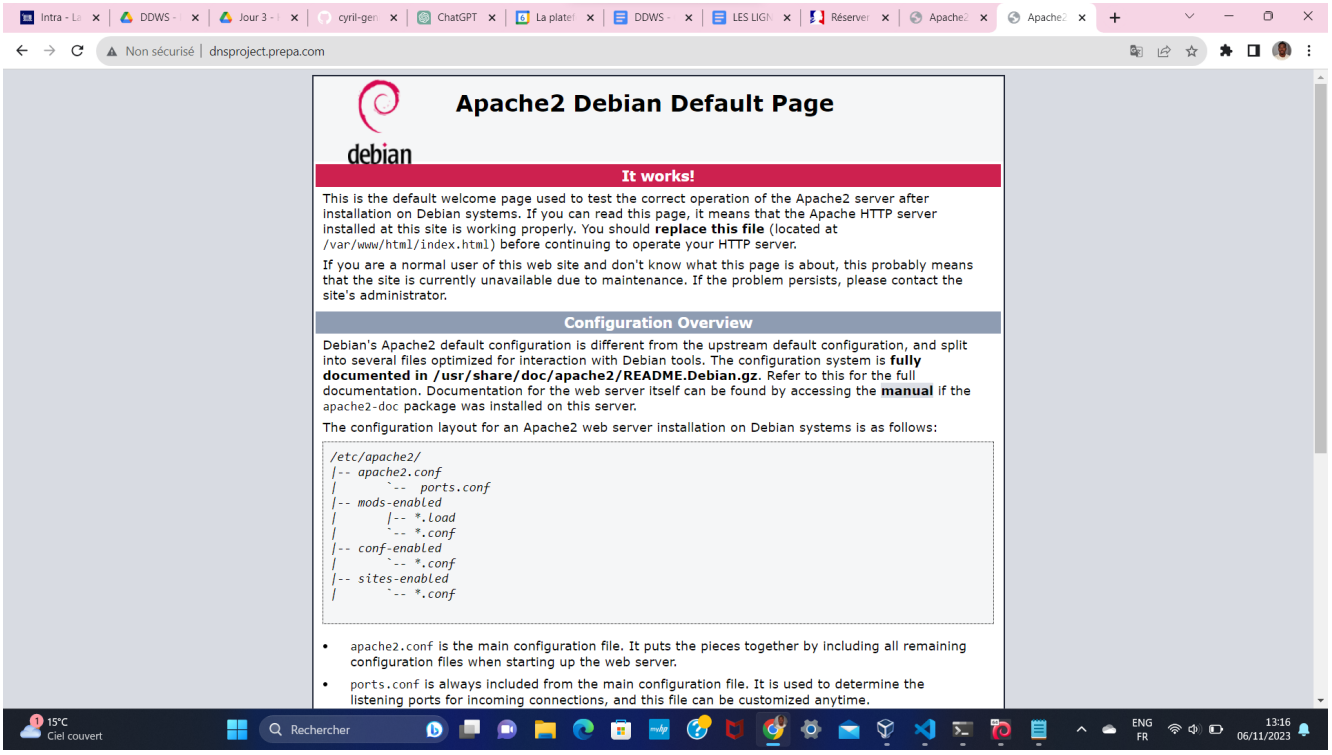
  named-checkzone /etc/bind/db.dnsproject.prepa.com

JOB 5

- Le nom de domaine permet aux internautes d'identifier votre site internet et d'accéder à ses pages web. Il peut ainsi être judicieux de réserver un nom de domaine pour communiquer sur votre activité, rallier une clientèle et élargir votre réputation commerciale. On obtient un nom de domaine par l'intermédiaire d'un bureau d'enrégistrement par exemple, sans faire de publicité GANDI. Ce bureau effectue alors la validation de la propriété pour un certain temps auprès d'un Register, une sorte de greffier des noms de domaine. Au niveau mondial, c'est l'ICANN (Internet Corporation for Assigned Names and Numbers) qui alloue l'espace d'adressage des protocoles internets. Cette société de droit californien délègue son autorité à des registrars nationaux. En France, il s'agit de l'Afnic.
- Pour certains domaines, il faut pouvoir justifier de sa nationalité, ou encore de sa résidence dans une ville particulière. Chaque extension de nom de domaine peut avoir ses propres contraintes.

JOB 6

Connectons notre hôte au nom de domaine local de notre serveur, pour que notre page apache soit accessible via ce même nom de domaine.



JOB 7

Mise en place d'un pare-feu utilisant **ufw (Uncomplicated Firewall)** sur notre serveur principale :

- Configuration du pare-feu. Pour ce faire il faut d'abord :

➢ Installer le pare-feu ufw avec la commande : **sudo apt install ufw**

➢ Configurer les règles de pare-feu. Nous bloquons tout le traffic entrant en le configurant par défaut mais nous autorisons seulement le traffic HTTP (port 80) et SSH (port 22) et HTTPS (port 443) en exécutant les commandes suivantes :

sudo ufw allow 80/tcp

sudo ufw allow 22/tcp

sudo ufw allow 443/tcp

- Activation de ufw avec la commande : **sudo ufw enable**
- Vérification de l'état de notre pare-feu : **sudo ufw status**
- Si nous voulons désactiver notre pare-feu, il suffit d'exécuter la commande : **sudo ufw desable.**

JOB 8

Pour cet exercice, supposons que tous les autres membres sont sous Linux. Si tel est le cas, alors nous allons mettre en place un serveur **NFS (Network File System)**

- Installation de NFS avec la commande : sudo apt install nfs-kernel-server
- Configuration des partages NFS :

➢ Création du répertoire que nous souhaitons partager. Personnellement j'ai nommé mon répertoire **"partage**". On exécute la commande suivante pour créer le répertoire : **sudo mkdir /partage**

➢ Édition du fichier **/etc/exports** pour spécifier les options de partage. Ajoutons une ligne avec les informations suivantes :

mkdir -p /srv/partage

/partage Adresse IP Client(rw,sync,no_subtree_check)

- **Adresse IP Client** remplace l'adresse IP du client auquel vous souhaitez autoriser l'accès.
- **rw** autorise la lecture et l'écriture

- **sync** synchronization synchrone est utilisé pour garantir que les données sont écrites sur le serveur avant de répondre aux clients).
- **no_subtree_check** désactive la vérification de sous-arbre. (passer cette option pour éviter les erreurs de montage liées à la vérification des sous-répertoires).

➢ Redémarrage de ufw : **sudo systemctl restart nfs-kernel-server**

➢ Autorisation du port NFS dans le pare-feu : **sudo ufw allow Adresse_IP_du client_NFS to any port nfs**