

JOB 1 :

La commande pour installer SSH est `sudo apt install openssh-server`

JOB 2 :

La commande pour installer apache2 est `sudo apt install apache2`

JOB 3 :

Apache est : Open-source et gratuit même pour un usage commercial, il a des mises à jour régulières, correctifs de sécurité réguliers, flexible grâce à sa structure basée sur des modules, Plateforme-Cross (fonctionne sur les serveurs Unix et Windows) et fonctionne avec les sites WordPress. Mais, il a des problèmes de performances sur les sites web avec un énorme trafic et il a trop d'options de configuration pouvant mener à la vulnérabilité de la sécurité. Nginx est plus rapide, il sert du contenu statique environ 2,5 fois plus rapidement qu'Apache et il gère mieux le trafic élevé qu'Apache. Mais il a : des options limitées, une communauté moins développée que celle d'Apache et une moins bonne option pour servir du contenu dynamique qu'Apache. Tomcat est facile à installer et simple à configurer, il possède des fonctionnalités de sécurité intégrées et il a des options simples pour le déploiement d'applications Web. Mais, il a : un problème lié à une fuite de mémoire et une prise en charge des clusters qui n'est pas

JOB 4 :

Pour pouvoir mettre en place un DNS sur notre serveur Linux qui fera correspondre l'adresse IP de notre serveur au nom de domaine local suivant est : `sudo nano /etc/hosts` et j'y écrit : 127.0.0.1
dnsproject.prepa.com

JOB 5 :

Un nom de domaine est l'adresse que les visiteurs vont renseigner afin d'accéder à votre site Internet. Les règles de réservation d'un nom de domaine varient selon la nature du site : - Domaines géographiques à vocation nationale, selon la localisation géographique de l'entreprise : .fr (France), .de (Allemagne), .it (Italie), .eu (Union européenne) - Domaines génériques, à vocation internationale : .com (pour les activités commerciales), .net (pour les entreprises), .org (pour les associations ou organisations non gouvernementales, etc.) Le nom de domaine est attribué à celui qui en demande la réservation en premier. C'est donc la règle du premier arrivé, premier servi qui prévaut. Pour réserver un nom de domaine, il faut s'adresser à l'organisme gestionnaire qui en a la

charge. JOB 6 Pour pouvoir mettre en place un serveur DNS, il faut installer plusieurs paquets. La commande pour

JOB 6 :

Pour pouvoir mettre en place un serveur DNS, il faut installer plusieurs paquets. La commande pour installer plusieurs paquets à la fois est `sudo apt -y install`. Les paquets nécessaires pour ce job sont : `bind9`, `bind9utils` et `dnsutils`. Ensuite, on crée une copie du fichier de configuration `db.local` et on la nomme `direct`. J'y écris mon hostname qui est `debian`, le nom de domaine ici `dnsproject.prepa.com` et l'IP de mon serveur, ici `10.10.31.68`. Si on souhaite modifier mon IP j'exécute la commande suivante : `sudo ifconfig nom_carte_reseau IP_souhaité`. Si la commande est introuvable on l'installe en faisant : `sudo apt install net-tools`.

Ensuite on crée le fichier inverse. Pour aller plus vite, on copie le fichier de configuration `direct` et l'appelle `inverse`.

Dans ce fichier on modifie uniquement la dernière ligne en mettant `Les_deux_derniers_octets_IP IN PTR Hostname.address_DNS`. On modifie le fichier `named.conf.local` pour qu'il prenne en compte les fichiers de configuration `direct` et `inverse`. Enfin le dernier fichier à modifier est le fichier `resolv.conf` pour que notre ordinateur puisse prendre en compte notre DNS. On redémarre le service `bind9` J'exécute la commande `nslookup www` pour vérifier si le serveur DNS y est bien associé. Comme tout est fonctionnel, on peut maintenant passer à la configuration du côté hôte. On va dans le panneau de configuration, puis dans Réseau et Internet, dans wifi, dans Propriétés. Enfin dans Protocole Internet version 4, on met l'adresse IP de notre serveur DNS dans la section DNS préféré. Pour avoir encore accès à internet on met l'IP `8.8.8.8` dans DNS Auxiliaire.

JOB 7 :

Pour pouvoir faire un serveur DHCP, il faut tout d'abord l'installer. La commande est `sudo apt install isc-dhcp-server`. Nous l'avons installé dans les jobs précédents. Le premier fichier à modifier est `/etc/default/isc-dhcp-server`. Dans ce dernier, on indique sur quelle carte réseau le serveur doit être configuré. Pour connaître mon ip et le nom de mes cartes réseaux, on tape `ip a`. Ensuite, l'autre fichier à modifier est `/etc/dhcp/dhcpd.conf` Dans ce dernier on va y indiquer le sous réseau, le netmask, le DNS, le gateway et les paramètres nécessaires au fonctionnement du sous réseau. Enfin, on redémarre le service

JOB 9 :

Pour pouvoir installer ufw, on tape la commande suivante `sudo apt install ufw`. L'installation a été faite dans les jobs précédents. La configuration initiale est de tout bloquer, puis d'ouvrir les ports selon nos besoins. Pour activer ufw, il faut taper `sudo ufw enable`. Pour le désactiver c'est `sudo ufw disable`. Pour pouvoir ajouter des règles à ufw, il faut taper `sudo ufw allow n°_port/tcp` Ici nos besoins sont : SSH (22) DNS (53), DHCP (67), HTTP/apache2 (80), HTTPS (443).

JOB 10 :

Pour pouvoir créer un dossier partagé, il faut installer samba. Pour cela, la commande est `sudo apt install samba` Puis on active le service On édite le fichier `/etc/samba/smb.conf`. On y met les paramètres du dossier : son nom, son chemin d'accès, les permissions, l'accès guest qui correspond à la connexion invité ou anonyme ou encore sa visibilité sur les autres ordinateurs. On redémarre le service : L'étape qui suit est d'ajouter un utilisateur a samba. Pour cela on va assigner un mot de passe à l'utilisateur par samba. La commande est : `sudo smbpasswd -a nom_utilisateur`. Ensuite il faut créer un groupe ayant un nom qui fait par exemple référence au nom du dossier On crée le dossier qui va être partagé et donne la propriété du dossier au groupe précédemment créé et on lui donne les droits d'écriture et de lecture. Pour finir n'oublions pas d'ajouter samba à ufw