

Job 1:

installation d'une VM debian avec interface graphique.

Job 2:

installation de apache avec la commande `sudo apt-get install apache2` puis taper son adresse ip pour accéder au site apache avec le message "it works!"

Job 3:

Qu'est-ce qu'un serveur web?

Les serveurs de fichiers, les serveurs de bases de données, les serveurs de messagerie et les serveurs web utilisent différents types de logiciels de serveur. Chacune de ces applications peut accéder aux fichiers stockés sur un serveur physique et les utiliser pour différents buts.

Le travail d'un serveur web consiste à servir des sites web sur internet. Pour atteindre cet objectif, il agit comme un intermédiaire entre le serveur et les machines des clients. Il extrait le contenu du serveur sur chaque requête d'utilisateur et le transmet au web.

Le plus grand défi d'un serveur web est de servir simultanément plusieurs et différents utilisateurs web – chacun demandant des pages différentes. Les serveurs web traitent les fichiers écrits dans différents langages de programmation tels que PHP, Python, Java et autres.

Ils les transforment en fichiers HTML statiques et diffusent ces fichiers dans le navigateur des utilisateurs web. Quand vous entendez le mot serveur web, considérez-le comme l'outil responsable de la communication serveur-client.

Apache:

Apache est l'un des serveurs web les plus populaires qui vous permet de gérer un site web sécurisé sans trop de problèmes. Apache est un serveur web, mais ce n'est pas un serveur physique mais plutôt un logiciel qui s'exécute sur un serveur. Son

travail consiste à établir une connexion entre un serveur et les navigateurs des visiteurs d'un site web (Firefox, Google Chrome, Safari, etc.) tout en délivrant des fichiers entre eux (structure client-serveur). Apache est un logiciel multiplateforme, il fonctionne donc à la fois sur les serveurs Unix et Windows.

Apache est hautement personnalisable, car il a une structure basée sur des modules. Les modules permettent aux administrateurs de serveur d'activer ou de désactiver des fonctionnalités supplémentaires. Apache possède des modules pour la sécurité, la mise en cache, la réécriture d'URL, l'authentification par mot de passe et encore plus. Vous pouvez également configurer vos propres configurations du serveur via un fichier appelé `httpd.conf` qui est un fichier de configuration Apache.

Apache possède beaucoup d'avantages comme le fait qu'il soit open source et gratuit donc tout le monde peut l'utiliser. Le logiciel est stable et fiable, facile à utiliser et configuré pour des débutants. Il est ce qu'on appelle plateforme cross, c'est à dire qu'il peut être utilisé sur une distribution debian ou windows. Il y a beaucoup de mises à jour régulières, et une communauté active qui peut aider en cas de problèmes.

Mais il y a aussi des inconvénients comme le fait qu'il y a beaucoup de configuration possible ce qui peut perdre les débutants. Ou alors qu'il peut y avoir des problèmes de performances sur les sites webs avec beaucoup de trafic.

Nginx

Nginx créé par Igor Sysoev est conçu pour offrir une faible utilisation de la mémoire et une grande simultanéité. Plutôt que de créer de nouveaux processus pour chaque requête Web, Nginx utilise une approche asynchrone et événementielle où les requêtes sont traitées dans un seul thread.

Avec Nginx, un processus maître peut contrôler plusieurs processus de travailleurs. Le maître gère les processus du travailleur, tandis que les travailleurs effectuent le traitement proprement dit. Comme Nginx est asynchrone, chaque requête peut être exécutée simultanément par le travailleur sans bloquer les autres requêtes.

Nginx est souvent comparé à Apache car les 2 applications sont beaucoup utilisées dans le milieu du web serveur. Il y a des choses où Nginx est meilleur que Apache dans certains domaines.

Il est open source et multiplateforme tout comme Apache. Lors de son utilisation il n'y a pas besoin de beaucoup de ressources alors que Apache lui en demande beaucoup et pourrait alors saturer alors que Nginx non.

Nginx est aussi fait pour accélérer le fonctionnement des serveurs. Cela n'est pas pris en charge par Apache.

Mais après il possède beaucoup de défaut que Apache n'a pas, comme le fait de pouvoir personnaliser son serveur comme on le souhaite avec beaucoup d'options et accessible au débutant. Et possède aussi un fichier de configuration nommé htaccess. Pour finir Apache possède un bon nombre de modules permettant d'étendre son serveur.

Tomcat:

Apache Tomcat est un logiciel de serveur d'applications web open source conçu pour la programmation en Java et développé et maintenu par Jakarta, le groupe de projets open source Java de la fondation Apache.

L'objectif initial du logiciel Apache Tomcat est d'héberger et de déployer les servlets Java.

Les servlets sont les programmes Java exécutés du côté serveur et qui reçoivent les requêtes des clients, les interprètent et génèrent les réponses demandées.

L'envoi des résultats peut se faire de façon directe ou via un protocole comme http.

Apache Tomcat fournit les fonctionnalités de base du traitement du serveur web pour les servlets Java.

Tout comme les webs serveurs précédents il est open source et gratuit. Il est mis à jour régulièrement. Il est très sécurisé à l'aide d'un protocole SSL. Il est léger et performant. Il est aussi multiplateforme. Mais par rapport aux 2 autres il est vraiment très limité, on le voit plus comme un support de Apache. C'est vraiment un petit web serveur qui ne donne pas accès à une configuration très poussée et ne propose pas plusieurs modules et est centré sur les scripts java.

Job 4:

Pour mettre en place un DNS sur notre serveur linux pour faire l'adresse ip du serveur au nom du domaine "dnsproject.prepa.com". Pour cela il faut utiliser la commande `sudo nano /etc/hosts` pour pouvoir ajouter l'adresse ip du serveur et le nom de domaine.

Job 5:

Pour obtenir un nom de domaine public il faut d'abord savoir s'il est disponible. Il faudra donc faire des recherches pour savoir si il est disponible ou non. En effet, enregistrer un nom de domaine protégé par un droit de propriété intellectuelle peut exposer le déposant à des procédures judiciaires. Dès que cela est fait, on peut enregistrer son nom de domaine. La personne devra choisir l'extension souhaitée

puis choisir son nom de domaine. Le nom de domaine comporte entre 3 et 63 caractères et ne comporte pas de majuscules. Et pour finir le nom de domaine est payant selon l'extension que l'on choisi on a le (.fr) qui est généralement moins cher que le (.com)

Quelles sont les spécificités que l'on peut avoir sur certaines extensions de nom de domaine ?

Il existe 2 types d'extensions de domaine géographique ou générique.

Les extensions génériques, aussi appelées gTLD (pour *generic Top Level Domain*), sont formées par trois lettres (ou plus dans certains cas) indiquant le type de contenu et l'objet du site : .com pour commercial, .net pour internet, .org pour des organisations, etc.

Ces extensions peuvent être libres comme .com ou .net ou bien réservées à des organisations (.org) ou des associations (.asso). Il existe une vingtaine d'extensions de ce type.

Les extensions géographiques, aussi appelées ccTLD (pour *country code Top Level Domain*), contiennent deux lettres ou plus et sont généralement associées à un pays : .fr pour la France, .es pour l'Espagne, .de pour l'Allemagne, etc.

Job 6:

Pour connecter son hôte(VM) au domaine local(Windows) il va falloir passer par un bridge. Entre la VM et Windows. Pour cela il faut mettre la VM en bridge avant de la lancer il faut aller dans paramètre et modifier la configuration NAT en Bridge. Après il faut passer sur Windows aller dans les dossiers et suivre ce chemin "C:\Windows\System32\drivers\etc". Et accéder au fichier host et rajouter dedans l'ip de la VM et le nom local dns project.prepa.com.

Puis enregistrer et tester sur internet l'ip et le nom local pour pouvoir se connecter à la page apache

Job 7:

Après une journée sans réussite et avec un pc me permettant pas de pouvoir supporter 1 VM. J'ai fait le travail avec quelqu'un sans réussite. Voilà ce qu'on a fait: on installe le serveur dhcp sudo apt-get install server-dhcp3. Ensuite on accède au

fichier interfaces à l'aide la commande nano/etc/network/interfaces qu'on a ouvert sur chaque VM et on a mis une ligne de code dedans iface eth0 inet dhcp(cette commande sert à attribuer une adresse IP automatiquement. Ensuite on accède à la VM serveur ou on ouvre la configuration du dhcp (nano/etc.dhcp3/dhcpd.conf). Quand on à ouvert le fichier on a rajouté le nom de domaine "dnsproject.prepa.com" le default-lease-time qui indique le temps de vie de l'adresse IP. Le Max-lease-time qui est le temps de vie maximum. Le Subnet(l'adresse réseau qu'on va utiliser) et le netmask(masque de sous réseau) et pour finir la Range qui est la plage d'adresse IP minimum et maximum donnée. Après on ferme le nano en sauvegardant et on démarre le serveur dhcp à l'aide de la commande /etc/init.d/dhcp3-server start. Ensuite sur la VM qui est reliée au serveur il faut redémarrer l'interface réseau /etc/init.d/networking restart et normalement ça devrait marcher.

Job 8:

Déjà pour ce job il faut passer les VM en mode bridge. Pour cela il faut lancer VMware clique droit sur sa VM puis modifier et passer de NAT en Bridge pour les 2 VM puis repasser dans le fichier nano/etc/.dhcp3/dhcpd.conf et rentrer la commande "options routeurs" et rentrer notre passerelle. puis enlever le wi-fi sur la VM qui n'est pas le serveur pour voir si ça se connecte au wi-fi.