En vous aidant d’internet, recherchez et répondez aux questions suivantes. Cela vous servira de cours sur Apache. Vous devrez savoir, par la suite, répondre à chacune de ces questions, car il s’agit de la base d’un serveur Web.

Quel paramètre rajouter à un Virtual host pour définir sur le site web une page d'ouverture par défaut possédant un nom différent de ceux utilisés d'habitude ?   
  
Pour définir une page d'ouverture par défaut avec un nom différent dans un virtual host Apache, vous pouvez utiliser la directive DirectoryIndex dans la configuration du virtual host. Par exemple :

<VirtualHost \*:80>

ServerAdmin webmaster@example.com

DocumentRoot /chemin/vers/le/site

DirectoryIndex ma\_page\_daccueil.html index.html index.php

<Directory "/chemin/vers/le/site">

Options FollowSymLinks

AllowOverride None

Require all granted

</Directory>

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

* Dans cet exemple, ma\_page\_daccueil.html est défini comme la première page à rechercher lors de l'accès au site. Vous pouvez ajuster l'ordre des noms de fichiers selon vos besoins.
* Comment définir la page par défaut qui s'affiche en cas d'erreur ?

Pour définir la page par défaut qui s'affiche en cas d'erreur dans un virtual host Apache, vous pouvez utiliser la directive ErrorDocument. Voici un exemple de configuration dans un fichier de virtual host :

<VirtualHost \*:80>

ServerAdmin webmaster@example.com

DocumentRoot /chemin/vers/le/site

<Directory "/chemin/vers/le/site">

Options FollowSymLinks

AllowOverride None

Require all granted

</Directory>

ErrorDocument 404 /erreur404.html

ErrorDocument 500 /erreur500.html

ErrorLog ${APACHE\_LOG\_DIR}/error.log

CustomLog ${APACHE\_LOG\_DIR}/access.log combined

</VirtualHost>

Dans cet exemple, si une erreur 404 (page non trouvée) se produit, Apache affichera le contenu de /erreur404.html. Vous pouvez également spécifier des pages personnalisées pour d'autres codes d'erreur en utilisant des directives ErrorDocument supplémentaires. Assurez-vous que les chemins spécifiés correspondent à la structure de votre serveur.

* Comment créer un site en https ?

1. Pour créer un site en HTTPS avec Apache, suivez ces étapes :

1. Installez Apache : Assurez-vous d'avoir Apache installé sur votre serveur. Vous pouvez l'installer en utilisant le gestionnaire de paquets de votre système d'exploitation (par exemple, apt, yum, brew).

2. Installez OpenSSL : Si ce n'est pas déjà fait, installez OpenSSL, un outil qui permet de générer des clés SSL.

3. Installez le module SSL d'Apache : Assurez-vous que le module SSL est activé. Vous pouvez le faire en exécutant la commande suivante :

sudo a2enmod ssl

4. Générez une paire de clés SSL : Utilisez OpenSSL pour générer une clé privée et une demande de certificat. Par exemple :

sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/votre\_domaine.key -out /etc/ssl/certs/votre\_domaine.crt

2. Suivez les instructions pour remplir les détails du certificat.

5. Configurez le virtual host : Modifiez la configuration de votre virtual host Apache pour activer le support SSL. Voici un exemple :```apache<VirtualHost \*:443> ServerName votresite.com DocumentRoot /chemin/vers/le/site

SSLEngine on SSLCertificateFile /etc/ssl/certs/votre\_domaine.crt SSLCertificateKeyFile /etc/ssl/private/votre\_domaine.key ErrorLog ${APACHE\_LOG\_DIR}/error.log CustomLog ${APACHE\_LOG\_DIR}/access.log combined </VirtualHost>

6. \*\*Redémarrez Apache :\*\* Redémarrez Apache pour appliquer les changements.

bash sudo service apache2 restart ```

7. Vérifiez votre site en HTTPS : Accédez à votre site en utilisant le protocole HTTPS (https://votresite.com/). Assurez-vous qu'aucun avertissement de sécurité n'apparaît dans le navigateur.

Ces étapes sont une configuration de base. Si vous avez un certificat SSL provenant d'une autorité de certification (CA), assurez-vous de suivre leurs instructions spécifiques pour l'installation.