

Programme

420.B0 Réseautique : Infonuagique et sécurité (DEC)

LEA.BP Réseaux infonuagiques (AEC)

Titre du cours :	Services Réseaux Ouverts
Code du cours :	420-SRO-TT
Pondération :	2-3-2



Configuration du service web



Enseignant : Miloud SETRA
Session : Automne 2023
Groupe : 531/532/533/537

Sommaire

- Configuration de base
- Publication web : dossier virtuel
- Publication web : dossier personnel des utilisateurs
- Publication web : hôtes http virtuels

I. Configuration de base du service WEB

1. INTRODUCTION

Le protocole HTTP

HTTP ou HyperText Transfer Protocol est un protocole de requêtes et de réponses. Le dialogue entre un client web (un navigateur tel que Netscape) et un serveur (Apache) se traduit par une requête du client à laquelle le serveur répond en effectuant le traitement intermédiaire adéquat.

Par exemple lorsque le navigateur transmet l'URL(Uniform Resource Locator) <http://www.teccart.tld/index.html> au serveur Web. Ce dernier commence par extraire la partie initiale, qui définit le protocole à utiliser (http), ainsi que le nom du serveur à contacter (www.teccart.tld). Ensuite il se connecte à cette machine en utilisant le port TCP approprié (80 dans le cas du protocole http).

Le protocole SSL (Secure Sockets Layer) (HTTPS)

Parfois, l'URL utilisée pour accéder à une page commence par https://. Cela signifie que la connexion se fait par le biais du protocole SSL.

Dans la pratique, cela correspond à plusieurs choses. D'abord, votre navigateur se connecte au serveur sur un port différent (443) au lieu du port http classique (80). Ensuite, toutes les informations transmises entre votre navigateur et le serveur sont chiffrées à l'aide d'un algorithme complexe qui vous protège contre toute interception de l'échange. Enfin, le serveur est identifié formellement, afin de vous protéger des imposteurs. Le protocole SSL n'est pas spécifique à HTTP.

2. INSTALLATION

Installer avec :

```
dnf install httpd
```

I. Lancer, arrêter et redémarrer le serveur

Pour démarrer le service :

```
# systemctl restart httpd
```

Pour activer le service :

```
# systemctl enable httpd
```

Pour arrêter / vérifier le service :

```
# systemctl stop/status httpd
```

:

Il existe également un outil nommé `apachectl` qui propose de lancer et d'arrêter le serveur plus facilement.

Démarrer le service :

```
# apachectl start
```

Arrêter le service :

```
# apachectl stop
```

Redémarrer le service :

```
# apachectl restart
```

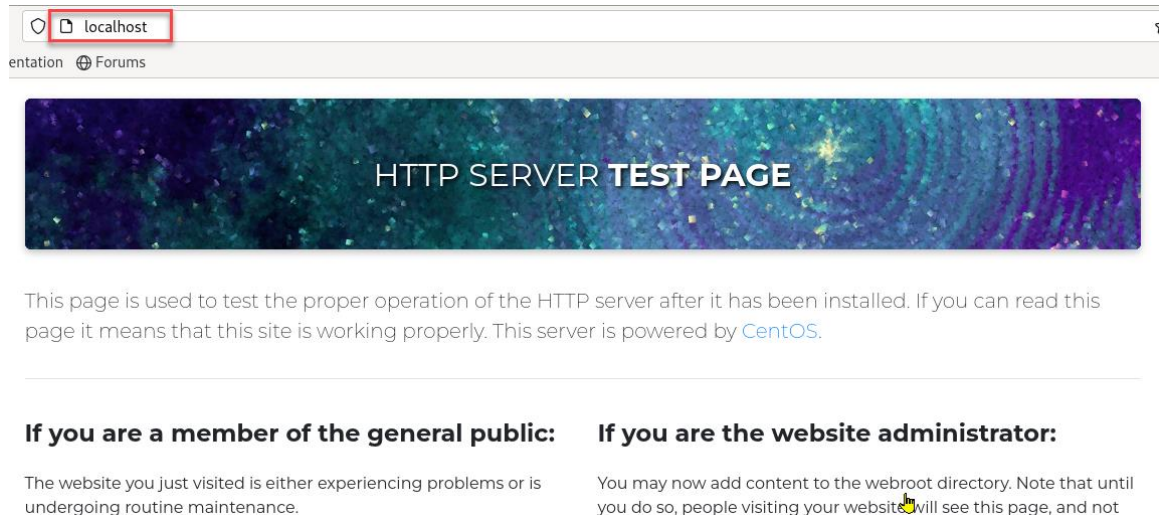
On peut activer Apache avec la commande `ntsysv` ou `setup` pour qu'il soit lancé au démarrage.

II. Tester le serveur Apache

Pour tester le bon fonctionnement du service Apache consiste à lancer un navigateur web et d'utiliser:

http://localhost

comme adresse. Après l'installation et le lancement du service Apache, vous devriez voir apparaître la page web par défaut d'Apache.



III. CONFIGURATION DE BASE DU SERVEUR HTTP

La configuration du serveur Apache peut s'effectuer soit en modifiant manuellement son fichier de configuration avec un éditeur de texte. Même si la configuration du service Apache à partir d'un fichier texte peut paraître fastidieuse, cette méthode présente l'avantage de contrôler exactement et de façon exhaustive le fonctionnement du serveur.

Le fichier de configuration du service Apache se nomme **httpd.conf** et est placé dans le répertoire conf/ du répertoire d'Apache. Dans le cas d'une installation à partir d'un paquetage, le fichier httpd.conf se trouve généralement dans le répertoire /etc/httpd/conf

Comment Apache structure-t-il sa configuration ?

- Les directives de conteneur Apache ont une portée limitée ;
- Les directives peuvent être utilisées à portée globale ou locale ;
- Il est possible d'outrepasser une directive à l'aide d'une configuration par répertoire.

Vous trouverez la liste complète de toutes les directives reconnues par le service Apache (version 2.4) à l'adresse :

<http://httpd.apache.org/docs/2.4/>

Les directives les plus importantes :

3.1 Directives globales du fichier /etc/httpd/conf/httpd.conf

a) Racine du serveur

Syntaxe : ServerRoot nom répertoire

Défaut : ServerRoot /etc/httpd

La directive ServerRoot définit le répertoire dans lequel se situe les fichiers de configuration, les fichiers log ainsi que les modules du serveur. Typiquement, ce répertoire contiendra les sous-répertoires conf/ et logs/. Les chemins d'accès relatifs pour d'autres fichiers de configuration seront considérés relativement à ce répertoire.

b) Fichier PID

Syntaxe : PidFile nom répertoire

Défaut : PidFile run/httpd.pid

Fichier dans lequel le serveur enregistre son le PID du service apache. Le chemin est relatif à ServerRoot.

c) Timeout

Syntaxe : Timeout durée en secondes

Défaut : Timeout 120

Durée d'attente en secondes avant que le serveur ne retourne une erreur.

d) Port d'écoute du service httpd

Syntaxe : Listen adresse IP : numéro de port

Défaut : Listen 80

Listen 80

numéro est un nombre compris entre 0 et 65535; certains numéros de ports (surtout en dessous de 1024) sont réservés pour des protocoles spécifiques. Une liste des ports prédéfinis est consultable dans /etc/services; le port standard assigné au protocole HTTP est le port 80.

La directive Listen définit le port réseau que le serveur écoute. Le Port 80 est l'un des ports prédéfinis de Linux. Tous les ports numérotés en dessous de 1024 sont réservés à un usage système, c'est-à-dire. Que des utilisateurs non privilégiés (non-root) ne peuvent les utiliser ; ces derniers peuvent par contre utiliser des ports de plus haut rang. Listen 192.168.13.11:80

e) Groupe

Syntaxe : Group groupe Unix

Défaut : **Group apache**

La directive Group définit le groupe dont les requêtes seront traitées par le serveur.

f) User

Syntaxe : User utilisateurUnix

Défaut : **User apache**

La directive User définit l'utilisateur associé au serveur.

3.2 Directives de configuration du serveur principal

(1) Adresse électronique de l'administrateur

Syntaxe : **ServerAdmin** adresseEMail

Défaut : **ServerAdmin** root@localhost

La directive ServerAdmin définit l'adresse e-mail que le serveur inclut dans tout message d'erreur retourné au client.

Il peut être utile de dédier une adresse réservée à cet usage, comme:

ServerAdmin webmaster@teccart.tld

(2) Nom de serveur

Syntaxe : **ServerName** nom de domaine entièrement qualifié:port La directive ServerName définit le nom d'hôte du serveur et le numéro de port que le serveur utilise que s'identifier lui-même.

(a) **ServerName** **localhost :80**

Ou

(b) **ServerName** **www.teccart.tld**

(i) Emplacement par défaut des pages html (Répertoire racine)

Syntaxe : **DocumentRoot** : Racine de l'hôte http par défaut

Défaut : **DocumentRoot** [/var/www/html](#)

Cette directive définit le répertoire racine à partir duquel httpd va distribuer les fichiers. Le serveur ajoute le chemin relatif mentionné dans l'URL présentée à cette racine pour établir le chemin complet jusqu'au document.

Un accès à <http://www.teccart.tld/index.html> se réfère au document index.html se trouvant à la racine du serveur web principal: </var/www/html/index.html>

(ii) UserDir

Syntaxe: **UserDir public_html**

Défaut: **UserDir disabled**

La directive **UserDir** définit le sous-répertoire contenu dans répertoire personnel de l'utilisateur qui servira de location de la page personnelle de l'utilisateur (~utilisateur).

Le paramètre qui suit **UserDir** peut prendre l'une des valeurs suivantes:

- Le nom du sous-répertoire pour la page personnelle.
- Le mot-clé **disabled**. Ce qui permet de désactiver la translation utilisateur répertoire.
- Le mot-clé **disabled** suivi de la liste des utilisateurs dont la translation utilisateur répertoire est désactivée.
- Le mot-clé **enabled** suivi de la liste des utilisateurs dont la translation utilisateur répertoire est activée.

Pour activer la publication web à partir des répertoires personnels des utilisateurs, utiliser l'option **UserDir** comme ci-dessous:

Exemple :

UserDir enabled

UserDir public_html

Créer le fichier index.html dans ce dossier pour faire le test. Exemple pour accéder aux publications Web de bob dans son répertoire personnel, il faut taper:

<http://www.teccart.tld/~bob/index.html>

sera dirigé vers le dossier de publication de bob dans le dossier public_html dans son répertoire personnel:

/home/bob/public_html

Remarque : le dossier public_html n'existe pas par défaut, il faut le créer et y créer la page web index.html (Voir exemple plus loin)

Remarque : il faut s'assurer que les comptes de service apache (compte : apache et groupe : apache) aient les **permissions d'accès (rx)** aux dossiers des répertoires personnels des utilisateurs. il faut modifier les permissions des dossiers personnels :

chmod g+rx,o+rx /home/*

(iii) DirectoryIndex

Syntaxe : **DirectoryIndex** fichier1 fichier2...

Défaut : **DirectoryIndex** index.html index.htm index.php index.html

Définit l'ordre de recherche de la page par défaut à afficher. **DirectoryIndex** index.htm index.html index.php index.html

(iv) Journal d'erreur par défaut

Syntaxe : ErrorLog nomfichier

Défaut : **ErrorLog** logs/error_log

Cette directive définit le nom du fichier dans lequel le serveur marque la trace des erreurs rencontrées (error_log).

(v) ErrorDocument

Syntaxe : ErrorDocument code d'erreur document

Dans l'éventualité d'un problème ou d'une erreur, Apache peut exécuter l'une des quatre actions suivantes :

1. sortie d'un message d'erreur simple standard
2. sortie d'un message personnalisé
3. redirection vers une URL locale pour traiter le problème (ou l'erreur)
4. redirection vers une URL externe pour traiter le problème (ou l'erreur)

La première option est celle par défaut, les options 2 à 4 seront obtenues en utilisant la directive **ErrorDocument**, suivi du code HTTP d'erreur et du message textuel d'erreur, ou une URL.

Messages dans ce contexte, commence par un guillemet simple ("), qui ne fait pas partie du message lui-même. Apache ajoutera souvent des informations complémentaires explicitant le problème (ou l'erreur).

L'URL peut débuter par un slash (/) pour des URL locales, ou être complètement qualifiées. Exemples:

ErrorDocument 500 <http://foo.example.com/cgi-bin/tester>

ErrorDocument 401 /subscription_info.html

ErrorDocument 403 "Pas d'accès au site web"

Exercices :

I) Serveur Web principal:

1. Configurez le serveur web principal : avec le nom : `www.nom_domaine.tld` (en utilisant votre nom de domaine)

Exemple: `www.teccart.tld`

2. Créez une page web simple : **index.html** dans la racine du serveur principal (**/var/www/html/**)

Avec un éditeur de texte créez une page nommé `index.html` dans le dossier : `/var/www/html`

Voici un exemple de création de page Web `index.html`:

```
<H1>Bienvenue sur www.teccart.tld ... </H1>
```

3. Configurez le serveur DNS avec votre nom de domaine ainsi que les enregistrements de ressources (`srv1.teccart.tld IN A 10.30.32.100` et un CNAME `www.teccart.tld` qui pointe vers `srv1`)
4. Faire pointer votre machine vers votre DNS dans **/etc/resolv.conf**

II) Accès Web aux répertoires personnels des utilisateurs :

1. Créer un usager `mario`
2. Se connecter avec le compte `mario`
3. Créer le répertoire **public_html** dans le répertoire personnel de l'utilisateur `mario` (`/home/mario`).
4. Modifier les permissions : **755 pour /home/mario** (avec `chmod`)
chmod 755 /home/mario
5. Créer le fichier **index.html** dans **public_html** :

Fichier `index.html`:

```
<center><H1>Bienvenue sur la page web de Mario sur /home/mario</H1></center>
```


6. Se connecter comme root et modifier le fichier de configuration de apache (**etc/httpd/conf.d/userdir.conf**) de façon à permettre la visualisation de la page Web de l'utilisateur mario.

Fichier de configuration : **/etc/httpd/conf.d/userdir.conf**

**Modifiez les options suivantes (EN GRAS) dans ce fichier:
etc/httpd/conf.d/userdir.conf**

```
#  
# See also: http://httpd.apache.org/docs/misc/FAQ.html#forbidden  
#  
<IfModule mod_userdir.c>  
  #  
  # UserDir is disabled by default since it can confirm the  
  # presence of a username on the system (depending on home  
  # directory permissions).  
  #  
  # UserDir disabled  
  #  
  # publier des pages web à partir de leur répertoire personnel  
  
  UserDir enable  
  UserDir public_html  
  
</IfModule>
```

7. Redémarrer le service apache.
8. Changer les droits d'accès pour permettre l'accès au répertoire personnel de mario afin d'accéder à la page Web de l'utilisateur mario.

chmod 755 /home/mario

9. Pour accéder à la page web: il faut utiliser l'URL suivante:

<http://localhost/~mario>

ou

<http://www.teccart.tld/~mario>

Exercices:

III) Préparer le site web principal

Requis : DNS configuré et Selinux désactivé.

1. Configurez le serveur web principal : avec le nom :
www.nom_domaine.tld (en utilisant votre nom de domaine : www.teccart.tld)
2. Configurez le serveur DNS avec votre nom de domaine ainsi que les enregistrements de ressources (vmnet2.teccart.tld IN A 10.30.32.100 et un CNAME www.teccart.tld qui pointe vers vmnet2)
3. Faire pointer votre machine vers votre DNS dans **/etc/resolv.conf**

IV) Accès Web aux répertoires personnels des utilisateurs :

Fichier de configuration : **/etc/httpd/conf.d/userdir.conf**

Modifiez les options suivantes (EN GRAS) dans ce fichier:
etc/httpd/conf.d/userdir.conf

```
# The path to the end user account 'public_html' directory must be
# accessible to the webserver userid. This usually means that
# ~userid must have permissions of 711, ~userid/public_html must
# have permissions of 755, and documents contained therein must be
# world-readable.
# Otherwise, the client will only receive a "403 Forbidden"
# message.
#
# See also: http://httpd.apache.org/docs/misc/FAQ.html#forbidden
#
<IfModule mod_userdir.c>
#
# UserDir is disabled by default since it can confirm the
# presence of a username on the system (depending on home
# directory permissions).
#
# UserDir disabled
#
# publier des pages web à partir de leur répertoire personnel

UserDir enabled
UserDir public_html

</IfModule>
```

4. Redémarrer le service apache.

Remarque : le dossier `public_html` n'existe pas par défaut, il faut le créer et y créer la page web `index.html` (Voir exemple plus loin)

5. Créer un compte mario
6. Se connecter avec le compte mario
7. Créer le répertoire **public_html** dans le répertoire personnel de l'utilisateur mario (`/home/mario`)
8. Modifier les permissions : **755 pour /home/mario** (avec `chmod`)

chmod 755 /home/mario

9. Créer le fichier **index.html** dans le dossier **public_html** :

Exemple de fichier **index.html**:

```
<center><H1>Bienvenue sur la page web de Mario sur /home/mario</H1></center>
```

10. Pour accéder à la page web: il faut utiliser l'URL suivante:

<http://localhost/~mario>

ou

<http://www.teccart.tld/~mario>

V) Questions

11. Est-il possible de permettre la publication Web pour root à partir de son répertoire personnel : `/root`
12. Est-il possible de permettre la publication Web pour certains comptes et l'interdire pour d'autres comptes ?

VI) Publication Web à partir d'un dossier personnel

13. On veut publier du contenu Web à partir du dossier : `/mnt/web/teccart`
14. Créer le dossier avec les permissions appropriées

15. Autoriser l'accès au dossier au niveau de la configuration du service web httpd dans le fichier `/etc/httpd/conf/httpd.conf` : ajouter la section suivante :

```
<Directory "/mnt/web/teccart">  
    AllowOverride None  
    # Allow open access:  
    Require all granted  
</Directory>
```

16. Créer un fichier `index.html` personnalisé dans ce dossier
17. Dans le fichier de configuration `httpd.conf` créer un alias pour ce dossier :

```
Alias /teccart /mnt/web/teccart
```

18. Redémarrer le service httpd
19. Faire le test avec :

```
firefox http://www.teccart.tld/teccart
```

II. Configuration des hôtes http virtuels

LES HÔTES VIRTUELS

Le mécanisme des sites virtuels permet à un serveur Web d'héberger de multiples sites. Les fournisseurs d'accès ont souvent recours à cette technique, car ils ne souhaitent pas gérer une machine différente pour chacun des domaines.

Le terme Hôte Virtuel se réfère à la technique qui consiste à maintenir plus d'un serveur sur une même machine, tout en étant différenciés par leur nom apparent. Par exemple, il est souvent préférable pour des compagnies se partageant un serveur Web d'avoir leurs propres domaines accessibles par www.teccart.tld et ftp.teccart.tld

Apache est l'un des premiers serveurs à supporter des hôtes virtuels basés sur l'adresse IP. Les versions 1.1 d'Apache ainsi que les plus récentes supportent les deux types d'hôtes virtuels, ceux basés sur le nom et ceux basés sur l'adresse IP (appelés tous deux vhosts). Ces hôtes virtuels sont aussi souvent appelés hôtes virtuels nommés ou hôtes virtuels non nommés.

1. HÔTES VIRTUELS BASÉS SUR ADRESSES IP

Comme l'indique la mention "basé sur une adresse IP", le serveur doit disposer d'une adresse IP différente pour chaque hôte virtuel basé sur une adresse IP. Une adresse IP est associée à chaque interface réseau de votre ordinateur. Et si vous installiez deux interfaces, chacune avec sa propre adresse IP et un nom de domaine approprié. Lorsqu'un internaute demande www.teccart.tld/index.html, sa requête arrive sur la première carte. Même si le serveur ne connaît pas le nom de domaine utilisé dans la requête, il connaît l'adresse IP associée et peut alors distinguer les deux sites virtuels.

Nous détenons là une solution, même si elle a ses limites. Il peut se révéler difficile d'installer 80 cartes réseau sur une même machine afin d'héberger 80 domaines différents.

Une solution plus simple existe. Une même interface réseau pouvant utiliser plusieurs adresses IP (IP aliasing), il est inutile d'installer plusieurs cartes. La commande `ifconfig` permet de définir ces adresses supplémentaires. Il faut avoir au moins de deux adresses IP (comme par exemple celle de l'interface `bridge` et celle de l'interface `vmnet2` ou `3`).

La définition d'adresses IP multiples n'est que la première étape. Elles doivent être associées à des noms de domaines valides. Vous devez ensuite vérifier que le domaine est opérationnel et reconnu, par exemple à l'aide de `ping`. Si tout se passe bien, vous pouvez poursuivre par la configuration du serveur Apache, en éditant le fichier `httpd.conf` et en y ajoutant une section `VirtualHost` contenant, au minimum, une directive `DocumentRoot`.

EXEMPLE PRATIQUE 1

On voudra héberger les deux sites suivants :

Utilisez les adresses IP assignées à chaque élève

www.teccart.tld avec l'adresse IP 192.168.100.1

info.teccart.tld avec l'adresse IP 172.16.100.1

- **Il faut avoir deux interfaces avec deux adresses IP:**

Il est recommandé de créer une autre interface réseau virtuelle que vous allez configurer dynamiquement par DHCP ou avec une adresse IP statique.

Vérifier que le réseau fonctionne bien :

ping 172.16.100.1

et

ping 192.168.100.1

- **Créer les dossiers de publication Web pour les hôtes virtuels:**

Pour le site 1 `www.teccart.tld`:

`mkdir /var/www/wwwsite`

y mettre une page web personnalisé : `index.html`

Pour le site 2 `info.teccart.tld`

`mkdir /var/www/infosite`

y mettre une page web personnalisé : `index.html`

- Configurer les deux hôtes HTTP virtuels dans le fichier httpd.conf

Il faut ensuite modifier le fichier:

geany **/etc/httpd/conf.d/httpd-vhosts.conf**

afin d'y ajouter les deux sections suivantes à la fin du fichier :

```
<VirtualHost 192.168.100.1>
ServerName www.teccart.tld
DocumentRoot /var/www/wwwsite
ServerAdmin admin@teccart.ca
</VirtualHost>
```

```
<VirtualHost 172.16.100.1>
ServerName info.teccart.tld
DocumentRoot /var/www/infosite
ServerAdmin admin@teccart.ca
</VirtualHost>
```

- Ajoutez deux CNAME dans le DNS pour la résolution des noms FQDN des deux hôtes virtuels:

Le serveur doit pouvoir résoudre www.teccart.tld et info.teccart.tld , il faut les ajouter dans le fichier de zone du **DNS** pour pouvoir faire la résolution.

Faire un test avant de démarrer le service apache :

ping www.teccart.tld

et

ping info.teccart.tld

- Redémarrer le service httpd et faire les tests:

Enfin il faut redémarrer le service apache pour que ces modifications de configuration soient prises en compte : **service httpd restart**

Pour tester les modifications. Utiliser l'URL <http://www.teccart.tld/> pour afficher le premier site Web et l'URL <http://info.teccart.tld/> pour afficher le deuxième site Web.

Remarque : Ne pas oublier de créer une page web (index.html) dans /var/www/wwwsite et /var/www/infosite respectivement

Conclusion :

2. HÔTES VIRTUELS NOMMÉS

Alors que l'approche sur les hôtes virtuels basés sur une adresse IP fonctionne très bien, ce n'est pas la solution la plus élégante, car, pour tout hôte virtuel une adresse IP dédiée est nécessaire et il est difficile d'en obtenir sur certaines machines. Le protocole **HTTP/1.1** définit une méthode d'identification de serveur, par laquelle il peut déterminer à quel nom la requête a été adressée. La version 1.1 du service apache et les suivantes supportent cette technique aussi bien que la méthode traditionnelle de l'adresse IP pour chaque nom d'hôte.

L'avantage d'utiliser les hôtes virtuels nommés est d'avoir pratiquement un nombre illimité de serveurs, faciles à utiliser et à configurer. Il ne requiert aucun matériel ni logiciel supplémentaire. Le principal désavantage est que le client doit supporter cette partie du protocole. Les dernières versions de la plupart des navigateurs le font, mais il existe des anciens navigateurs encore en service qui ne le font pas.

EXEMPLE PRATIQUE 2

La différence notable entre la configuration des hôtes virtuels basés sur une adresse IP et celle des hôtes nommés est la directive **NameVirtualHost**, qui spécifie une adresse IP qui doit être utilisée comme une cible pour les hôtes virtuels nommés. De plus, toute directive supplémentaire peut et doit être placée à l'intérieur de la section **<VirtualHost>**.

Vous allez utiliser une seule adresse IP pour trois hôtes HTTP virtuels :

www.teccart.tld
info.teccart.tld
data.teccart.tld

- Dans le fichier `/etc/httpd/conf.d/httpd-vhosts.conf` :

Modifier/ajouter ce qui suit :

```
<VirtualHost 172.16.100.1>
ServerName www.teccart.tld
DocumentRoot /var/www/wwwsite
</VirtualHost>

<VirtualHost 172.16.100.1>
ServerName info.teccart.tld
DocumentRoot /var/www/infosite
</VirtualHost>

<VirtualHost 172.16.100.1>
ServerName data.teccart.tld
DocumentRoot /var/www/datasite
</VirtualHost>
```

- Créer le dossier de publication Web pour le 3eme site : **data.teccart.tld**:

mkdir /var/www/data

et y ajouter une page Web personnalisée **index.html**.

- **Modifier la configuration du DNS:**

Il faut ajouter 3 CNAME qui pointent vers le même enregistrement A appropriés dans (teccart.tld.db):

www	IN	CNAME	vmnet2
info	IN	CNAME	vmnet2
data	IN	CNAME	vmnet2

Redémarrer le service named et faire les tests avec dig pour s'assurer de la résolution des trois CNAME ci-dessus.

- **Créer une page web index.html personnalisée pour chaque configuration**
- **Faire le test avec lynx pour accéder aux trois sites http virtuels**
 - Lynx www.teccart.tld puis info.teccart.tld et enfin data.teccart.tld