

Rapport

KAIDI Ahmed El Aziz
RHARMAOUI Rafik
ENNAJI WASSIM

April 12, 2024

Projet de Système Et Réseaux II

1 Introduction

Au cours de ces travaux pratiques, nous nous attelons à la mise en place d'un réseau d'entreprise et à l'élaboration d'une infrastructure de services, reproduisant ainsi un environnement opérationnel réaliste. Notre groupe, composé de trois étudiants, jouera le rôle d'une agence au sein de cette entreprise, étant connecté aux autres agences par le biais d'un réseau dédié. Nous disposons à cet effet d'un serveur HPE, utilisé comme routeur, ainsi que de deux ordinateurs portables Dell, appelés « clients ».

Notre objectif est de suivre un ensemble d'étapes définies afin de configurer le réseau conformément à une topologie préétablie. Ces étapes incluent notamment l'installation d'une distribution GNU/Linux Debian 12 Bookworm, sans interface graphique, via le réseau en utilisant le protocole PXE. De plus, nous devons ajuster les paramètres réseau pour attribuer des adresses IP fixes, mettre en place les réseaux d'interconnexion et privé pour notre agence, et activer le routage entre ces réseaux.

Il est crucial de noter que toute modification de configuration apportée aux machines temporaires doit être annulée avant la fin des travaux pratiques. De plus, nous devons fournir un compte rendu détaillé de nos opérations, comprenant une description minutieuse de chaque étape de la configuration et des protocoles expérimentaux employés pour les valider.

En collaboration avec les autres groupes, nous devons également élaborer un plan d'adressage cohérent afin d'assurer la connectivité entre les différentes agences. À l'issue de ces travaux pratiques, nous devons soumettre un rapport complet, exposant nos opérations et les protocoles utilisés, ainsi qu'une synthèse des commandes du gestionnaire de paquets Debian, essentielles pour administrer efficacement un système GNU/Linux Debian.

Contents

1	Introduction	1
2	Installation d'une distribution GNU / Linux Debian 12	4
3	Partition et formatage du disque dur	4
4	Plan Adressage et routage	5
4.1	Explication détaillée de la configuration réseau dans le fichier <code>interfaces</code>	5
5	Test des interfaces réseau :	8
6	Table de routage :	9
7	Configuration IPTABLE pour accéder à Internet :	9
8	Implémentation du service DHCP (isc-dhcp-server) sur le routeur :	10
8.1	Installation du service DHCP :	10
8.2	Configuration de l'interface d'écoute :	10
8.3	Configuration du serveur DHCP :	10
8.4	Redémarrage du service DHCP :	10
9	Mise en place d'une sauvegarde avec rsync et cron :	12
10	Synthèse des commandes du gestionnaire de packages Debian :	13
11	Conclusion TP1	14
12	Introduction TP 1bis	14
13	Manuel Des Commandes (host, nslookup, dig)	14
14	Analyse de la résolution DNS pour le domaine <code>www.u-bourgogne.fr</code>	16
15	Analyse de la Réponse DNS et Changement de Serveur DNS	18
15.1	Identification du Serveur DNS Répondant	18
15.2	Changement de Serveur DNS avec dig	18
16	Analyse des Serveurs de Noms pour le Domaine "u-bourgogne.fr"	18
16.1	Résultats de la Commande <code>host -t ns u-bourgogne.fr</code>	18
16.2	Résultats de la Commande <code>dig u-bourgogne.fr ns</code>	18
16.3	Analyse Comparative	18
17	Consultation DNS et Suivi de la Trace avec la Commande dig	19
17.1	Interrogation des Domaines avec la Commande dig	19
17.2	Observation du Résultat avec l'Option <code>+trace</code>	19
17.3	Analyse des Résultats	19
18	Gestion des Serveurs Racines DNS	19
19	Gestion de la Configuration du Serveur DNS	20
19.1	Gestion de la Configuration du Serveur DNS sur un Système Linux	20
19.2	Consultation de la Configuration Actuelle	20
19.3	Modification de la Configuration DNS	20

20 Implémentation et Configuration Personnalisée du Fichier /etc/hosts	20
20.1 Personnalisation du Fichier /etc/hosts	20
20.2 Utilisation en Développement Local	21
20.3 Blocage de Sites Web Indésirables	21
20.4 Gestion Prudente du Fichier /etc/hosts	21
21 Conclusion TP1bis	21
22 Introduction TP2	21
23 Installation d'un serveur HTTP	22
24 Création d'un compte utilisateur web1	22
25 Création d'un compte utilisateur web2	23
26 Création d'un compte utilisateur web3	24
27 Installation de Gparted et création d'une nouvelle partition	24
28 Installation de MariaDB	25
29 Installation et validation de PHP	26
30 Installation de PostgreSQL	27
31 Utilisation de PDO en PHP pour accéder et afficher les données des bases de données MySQL et PostgreSQL	28
32 Mise en place de la sauvegarde régulière des bases de données MySQL et PostgreSQL	29
33 Mise en place des règles de filtrage IPTables	30
34 Automatisation du routage et de la translation d'adresses au démarrage du serveur	31
35 Autorisation spécifique pour le webmaster : Modification de la configuration d'Apache et relance du serveur avec sudo	31
36 Conclusion TP2	32
37 Introduction TP 3	32
38 Définition du DIT (Directory Information Tree)	32
39 Installation des paquets et des schémas LDAP	33
40 Paramétrage du service LDAP	33
41 Paramétrage de l'exploitation des données de l'annuaire LDAP	34
42 Installation et configuration des smbldap-tools	34
43 Création des ressources à partager et configuration du service Samba	34
44 Peuplement de l'annuaire avec smbldap-populate	35
45 Conclusion TP3	35

2 Installation d'une distribution GNU / Linux Debian 12

Dans cette partie, nous décrivons le processus d'installation d'une distribution GNU/Linux Debian 12 en utilisant la fonction iLO (Integrated Lights-Out) de nos serveurs HPE. Cette méthode nous permettra d'installer Debian 12 à distance, offrant ainsi une flexibilité accrue dans la gestion de notre infrastructure réseau.

Le processus d'installation avec iLO peut varier selon la configuration spécifique de nos serveurs HPE, mais il suit généralement ces étapes :

Accès à iLO :

Nous accédons à l'interface iLO de notre serveur HPE en utilisant un navigateur Web et en entrant l'adresse IP appropriée.

Authentification :

Nous nous connectons à l'interface iLO en utilisant nos identifiants d'administration.

Montage de l'image ISO :

Nous montons l'image ISO de Debian 12 dans l'interface iLO, permettant ainsi au serveur de démarrer à partir de cette image.

Démarrage distant :

Nous redémarrons le serveur à distance en sélectionnant l'option de démarrage à partir de l'image ISO montée via iLO.

Installation de Debian 12 :

Nous suivons les étapes standard d'installation de Debian 12, en choisissant les paramètres appropriés tels que la langue, le clavier, etc., comme décrits dans la réponse initiale. Une fois cette procédure terminée, Debian 12 sera installé sur notre serveur HPE via iLO, nous permettant ainsi de procéder à la configuration ultérieure de notre infrastructure réseau.

Il est crucial de noter que la procédure d'installation avec iLO peut nécessiter une certaine familiarité avec l'interface iLO et les fonctionnalités spécifiques de nos serveurs HPE. Cependant, une fois maîtrisée, cette méthode offre un moyen pratique et efficace d'installer des systèmes d'exploitation à distance.

Cette étape initiale est essentielle pour établir une base solide pour la configuration ultérieure du réseau et de l'infrastructure de services, et l'utilisation de iLO facilite grandement ce processus en nous permettant de gérer nos serveurs à distance avec précision et efficacité.

3 Partition et formatage du disque dur

Lors de l'installation de GNU/Linux Debian, il est essentiel de partitionner et formater le disque dur de manière appropriée pour garantir le bon fonctionnement du système d'exploitation. Les partitions doivent être configurées avec soin pour répondre aux besoins spécifiques du système et des utilisateurs.

1. EFI :

La partition EFI (Extensible Firmware Interface) est nécessaire sur les systèmes UEFI pour stocker les fichiers de démarrage du système d'exploitation. Elle contient les fichiers de démarrage nécessaires pour lancer le système d'exploitation.

2. Racine (/) :

La partition racine contient le système d'exploitation lui-même. Toutes les applications système et les fichiers essentiels sont stockés dans cette partition. Elle est essentielle au fonctionnement de Debian.

3. Home (/home) :

La partition home est l'espace où les données des utilisateurs sont stockées. Cela inclut les documents, les fichiers personnels, les paramètres de configuration utilisateur, etc. Séparer cette partition du reste du système permet de préserver les données des utilisateurs en cas de réinstallation du système d'exploitation.

4. Swap :

La partition swap est utilisée comme espace de travail supplémentaire pour le système d'exploitation lorsque la mémoire vive (RAM) est pleine. Elle agit comme une extension de la mémoire RAM et permet de gérer efficacement la mémoire virtuelle du système.

5. Var/log :

La partition var/log est utilisée pour stocker les fichiers journaux système. Ces fichiers contiennent des informations sur les événements système, les erreurs, les avertissements, etc. Séparer cette partition du reste du système permet de prévenir les problèmes liés à un remplissage excessif du système de fichiers racine.

La configuration appropriée des partitions est essentielle pour assurer la stabilité, la performance et la sécurité du système d'exploitation GNU/Linux Debian. Elle permet également une gestion efficace de l'espace de stockage et des données utilisateur.

```
root@agence:/# lsblk
NAME        MAJ:MIN RM   SIZE RO TYPE MOUNTPOINTS
sda          8:0    0 447,1G  0 disk
├─sda1       8:1    0  93,1G  0 part /
├─sda2       8:2    0   954M  0 part /boot/efi
├─sda3       8:3    0    9,3G  0 part /home
├─sda4       8:4    0    9,3G  0 part /var
├─sda5       8:5    0  48,9G  0 part /BD
└─sda6       8:6    0    3,7G  0 part [SWAP]
root@agence:/#
```

Figure 1: les partitions sur un système GNU/Linux Debian

4 Plan Adressage et routage

4.1 Explication détaillée de la configuration réseau dans le fichier interfaces

Lors de la configuration réseau d'un système Debian, le fichier `/etc/network/interfaces` joue un rôle essentiel en définissant les paramètres réseau de manière précise. Ci-dessous, une analyse approfondie de chaque directive présente dans ce fichier :

1. **Inclusion des fichiers de configuration :** La directive `source /etc/network/interfaces.d/*` permet d'inclure tous les fichiers de configuration présents dans le répertoire `/etc/network/interfaces.d/`. Cette approche modulaire offre une gestion flexible et organisée de la configuration réseau.
2. **Configuration de l'interface loopback (lo) :** La directive `auto lo` configure automatiquement l'interface loopback (lo), une interface virtuelle permettant au système de communiquer avec lui-même. Cette configuration est indispensable pour les communications internes du système.
3. **Configuration de l'interface principale (ens1f0) :** La directive `allow-hotplug ens1f0` permet à Debian de gérer dynamiquement l'état de l'interface `ens1f0` en fonction du branchement ou du débranchement d'un câble réseau. Cette gestion automatique améliore la réactivité du système aux changements de connectivité réseau.

La directive `iface ens1f0 inet static` configure l'interface `ens1f0` en mode statique (`inet static`), ce qui implique une attribution manuelle des adresses IP. Cette approche est cruciale pour garantir une connectivité réseau stable et prévisible.

La directive `address 172.31.20.123/24` définit l'adresse IP statique de l'interface `ens1f0`, avec un masque de sous-réseau `/24`. Cette adresse permet au système de communiquer sur le réseau local de manière identifiable et constante.

La directive `gateway 172.31.20.1` spécifie l'adresse IP de la passerelle par défaut, utilisée pour router le trafic réseau vers des destinations externes au réseau local. La passerelle par défaut est un élément crucial de l'infrastructure réseau.

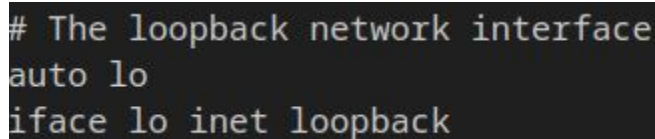
4. **Configuration des serveurs DNS :** La directive `dns-nameservers 172.31.21.35` configure les adresses IP des serveurs DNS à utiliser pour la résolution des noms de domaine en adresses IP. Cette configuration permet au système de traduire les noms de domaine en adresses IP, facilitant ainsi la navigation sur Internet et l'accès aux services réseau.
5. **Définition du domaine DNS :** La directive `dns-domain IEM` définit le domaine DNS à utiliser pour la résolution des noms de domaine locaux. Ce paramètre permet au système de résoudre les noms de domaine internes en adresses IP, favorisant ainsi la communication au sein du réseau local.

En conclusion, la configuration réseau dans le fichier `/etc/network/interfaces` fournit une base solide pour assurer une connectivité stable et fiable du système Debian. Chaque directive est soigneusement configurée pour répondre aux besoins spécifiques du réseau, garantissant ainsi un fonctionnement optimal dans divers environnements réseau.

Interface de bouclage (Loopback Interface)

```
auto lo
iface lo inet loopback
```

- Cette section configure l'interface de bouclage (`lo`).
- L'adresse IP `127.0.0.1` est utilisée pour référencer la machine elle-même.
- Cette interface est souvent utilisée pour les communications internes, telles que l'accès aux services locaux.



```
# The loopback network interface
auto lo
iface lo inet loopback
```

Figure 2: lookback

Interface IEM (IEM Interface)

```
allow-hotplug ens1f0
iface ens1f0 inet static
    address 172.31.20.123/24
    gateway 172.31.20.1
    dns-nameservers 172.31.20.123
    dns-domain IEM
```

- Cette section configure l'interface `ens1f0` pour le réseau IEM.
- L'adresse IP est définie sur `172.31.20.123` avec un masque de sous-réseau `/24`, ce qui signifie que le réseau local comprend les adresses IP de `172.31.20.0` à `172.31.20.255`.
- La passerelle par défaut (`gateway`) est définie sur `172.31.20.1`, indiquant le routeur ou le point de sortie vers d'autres réseaux.
- Les serveurs DNS sont définis sur `172.31.20.123` avec le domaine DNS "IEM".

```
# IEM
allow-hotplug ens1f0
iface ens1f0 inet static
    address 172.31.20.123/24
    gateway 172.31.20.1
```

Figure 3: IEM

Interface d'Interconnexion (Interconnexion Interface)

```
allow-hotplug ens1f1
iface ens1f1 inet static
    address 192.168.0.9/27
```

- Cette section configure l'interface **ens1f1** pour l'interconnexion.
- L'adresse IP est définie sur 192.168.0.9 avec un masque de sous-réseau /27, ce qui signifie que le réseau local comprend les adresses IP de 192.168.0.0 à 192.168.0.31.
- Le /27 indique que les 27 premiers bits sont réservés pour l'adresse réseau et les 5 bits suivants (32 - 27) sont réservés pour les adresses hôtes, fournissant ainsi 32 adresses IP disponibles dans le sous-réseau.

```
#Interco
allow-hotplug ens1f1
iface ens1f1 inet static
    address 192.168.0.9/27
```

Figure 4: interco

Interface Privée

```
allow-hotplug ens1f2
iface ens1f2 inet static
    address 10.2.9.1/24
```

- Cette section configure l'interface **ens1f2** pour le réseau privé.
- L'adresse IP est définie sur 10.2.9.1 avec un masque de sous-réseau /24, ce qui signifie que le réseau local comprend les adresses IP de 10.2.9.0 à 10.2.9.255.
- Le /24 indique que les 24 premiers bits sont réservés pour l'adresse réseau et les 8 bits suivants (32 - 24) sont réservés pour les adresses hôtes, fournissant ainsi 256 adresses IP disponibles dans le sous-réseau.

```
#Privee
allow-hotplug ens1f2
iface ens1f2 inet static
    address 10.2.9.1/24
```

Figure 5: Privee

5 Test des interfaces réseau :

Après avoir configuré les interfaces réseau pour le réseau d'interconnexion et le réseau privé de l'agence, nous avons procédé à une série de tests pour vérifier la connectivité et la fonctionnalité des réseaux.

Vérification des configurations :

- Nous avons utilisé la commande `ip addr show` pour afficher les informations sur les adresses IP attribuées à chaque interface. Toutes les configurations étaient correctes, avec les bonnes adresses IP et masques de sous-réseau.

Test de connectivité :

- Nous avons utilisé l'outil `ping` pour tester la connectivité entre les périphériques connectés à chaque réseau. Les tests de ping ont confirmé que les périphériques pouvaient se joindre les uns aux autres avec succès, tant localement que sur d'autres réseaux.
- La connectivité avec les passerelles par défaut a également été vérifiée en utilisant la commande `ping` avec l'adresse IP de la passerelle.

Test de résolution DNS (le cas échéant) :

- Si des serveurs DNS étaient configurés, nous avons utilisé la commande `nslookup` pour vérifier que la résolution DNS fonctionnait correctement. Les tests ont confirmé que les noms de domaine pouvaient être résolus avec succès en adresses IP.

Test de connectivité avec d'autres services :

- Nous avons également testé la connectivité avec d'autres services disponibles sur les réseaux, tels que des serveurs web ou des services de base de données. Les tests ont confirmé que les périphériques pouvaient accéder aux services disponibles sur d'autres périphériques du réseau.

En conclusion, les tests ont démontré que les configurations d'interfaces réseau étaient correctes et que les réseaux d'interconnexion et privé de l'agence étaient opérationnels, permettant une communication efficace entre les périphériques selon les exigences spécifiées.

```
root@AgenceLyon:~# ping 192.168.0.1
PING 192.168.0.1 (192.168.0.1) 56(84) bytes of data.
64 bytes from 192.168.0.1: icmp_seq=1 ttl=64 time=0.224 ms
64 bytes from 192.168.0.1: icmp_seq=2 ttl=64 time=0.233 ms
64 bytes from 192.168.0.1: icmp_seq=3 ttl=64 time=0.234 ms
64 bytes from 192.168.0.1: icmp_seq=4 ttl=64 time=0.239 ms
64 bytes from 192.168.0.1: icmp_seq=5 ttl=64 time=0.238 ms
64 bytes from 192.168.0.1: icmp_seq=6 ttl=64 time=0.235 ms
```

Figure 6: test interco


```

C:\Users\admin>ping 10.2.9.7

Envoi d'une requête 'Ping' 10.2.9.7 avec 32 octets de données :
Réponse de 10.2.9.7 : octets=32 temps<1ms TTL=128
Réponse de 10.2.9.7 : octets=32 temps<1ms TTL=128
Réponse de 10.2.9.7 : octets=32 temps<1ms TTL=128
Réponse de 10.2.9.7 : octets=32 temps<1ms TTL=128

Statistiques Ping pour 10.2.9.7:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 0ms, Maximum = 0ms, Moyenne = 0ms

```

Figure 7: test privée

6 Table de routage :

Voici les règles de routage actuellement configurées :

```

# Route par défaut vers la passerelle 172.31.20.1
ip route add default via 172.31.20.1

# Routes spécifiques vers d'autres réseaux
ip route add 10.2.6.0/24 via 192.168.0.6
ip route add 10.2.7.0/24 via 192.168.0.7
ip route add 10.2.8.0/24 via 192.168.0.8
ip route add 10.2.10.0/24 via 192.168.0.10
ip route add 10.1.1.0/24 via 192.168.0.1

```

Explications :

1. **Route par défaut vers la passerelle 172.31.20.1** : Cette règle spécifie que tout trafic destiné à des destinations inconnues doit être envoyé à la passerelle 172.31.20.1, qui agit comme la passerelle par défaut pour les réseaux externes.
2. **Routes spécifiques vers d'autres réseaux** : Ces règles spécifient des chemins de routage pour des sous-réseaux spécifiques :
 - Les adresses IP commençant par 10.2.6.0/24 seront acheminées via la passerelle 192.168.0.6.
 - Les adresses IP commençant par 10.2.7.0/24 seront acheminées via la passerelle 192.168.0.7.
 - Les adresses IP commençant par 10.2.8.0/24 seront acheminées via la passerelle 192.168.0.8.
 - Les adresses IP commençant par 10.2.10.0/24 seront acheminées via la passerelle 192.168.0.10.
 - Les adresses IP commençant par 10.1.1.0/24 seront acheminées via la passerelle 192.168.0.1.

Ces règles de routage assurent que le trafic est dirigé vers les passerelles appropriées en fonction de l'adresse de destination, permettant ainsi une connectivité entre les réseaux locaux et d'autres réseaux externes.

7 Configuration IPTABLE pour accéder à Internet :

Pour permettre à une machine du réseau privé (par exemple, 10.2.9.0/24) d'accéder à Internet, voici la règle IPTABLES appropriée :

```

# IPTABLE
up iptables -t nat -A POSTROUTING -s 10.2.9.0/24 -j MASQUERADE

```

Explications :

- Cette règle utilise la table de NAT (`-t nat`) pour traiter les paquets sortants.
- L'option `-s` spécifie la source des paquets, qui est le réseau privé (10.2.9.0/24).
- L'action `MASQUERADE` modifie l'adresse source des paquets sortants pour qu'ils apparaissent comme s'ils provenaient du routeur/firewall, permettant ainsi à plusieurs machines du réseau privé de partager une seule adresse IP publique.

Configuration au démarrage :

Pour que cette règle soit appliquée automatiquement au démarrage du système, vous pouvez l'ajouter au script d'initialisation des règles IPTables. Assurez-vous également d'enregistrer les règles IPTables pour qu'elles soient restaurées après le redémarrage.

8 Implémentation du service DHCP (isc-dhcp-server) sur le routeur :

Pour mettre en place le service DHCP (isc-dhcp-server) sur le routeur afin de fournir des adresses IP aux machines du réseau privé, suivez ces étapes :

8.1 Installation du service DHCP :

Tout d'abord, assurez-vous que le paquet `isc-dhcp-server` est installé sur le routeur en exécutant la commande suivante :

```
sudo apt-get install isc-dhcp-server
```

8.2 Configuration de l'interface d'écoute :

Ouvrez le fichier de configuration `/etc/default/isc-dhcp-server` avec un éditeur de texte :

```
sudo nano /etc/default/isc-dhcp-server
```

Assurez-vous que la variable `INTERFACESv4` est définie avec le nom de l'interface réseau correspondant au réseau privé. Par exemple :

```
INTERFACESv4="ens1f2"
```

Remplacez `ens1f2` par le nom de votre interface réseau.

8.3 Configuration du serveur DHCP :

Modifiez le fichier de configuration principal du serveur DHCP `/etc/dhcp/dhcpd.conf`. Voici un exemple de configuration de base pour attribuer des adresses IP aux clients du réseau privé :

```
subnet 10.2.9.0 netmask 255.255.255.0 {  
    range 10.2.9.100 10.2.9.200;  
    option routers 10.2.9.1;  
    option domain-name-servers 8.8.8.8;  
}
```

8.4 Redémarrage du service DHCP :

Une fois la configuration effectuée, redémarrez le service DHCP pour appliquer les changements :

```
sudo systemctl restart isc-dhcp-server
```

Assurez-vous de tester le service DHCP en connectant un client au réseau privé et en vérifiant s'il reçoit correctement une adresse IP attribuée par le serveur DHCP.

Fichiers utiles à la configuration de DHCP :

Pour la configuration du service DHCP (Dynamic Host Configuration Protocol), plusieurs fichiers sont utilisés. Voici une liste des fichiers les plus importants et utiles :

1. **/etc/dhcp/dhcpd.conf** : Ce fichier est le fichier de configuration principal pour le serveur DHCP. Il contient les paramètres de configuration spécifiques au serveur DHCP, tels que les plages d'adresses IP à attribuer, les options de configuration pour les clients DHCP, les sous-réseaux à servir, etc.
2. **/etc/default/isc-dhcp-server** : Ce fichier contient les options de configuration globales pour le service DHCP. Vous pouvez y spécifier les interfaces réseau sur lesquelles le serveur DHCP écoutera les demandes DHCP, ainsi que d'autres options de configuration spécifiques au service.
3. **/var/lib/dhcp/dhcpd.leases** : Ce fichier contient les informations sur les baux DHCP actuels accordés par le serveur DHCP. Il indique quelles adresses IP sont attribuées à quels clients, combien de temps le bail est valide, etc. Ce fichier est utilisé par le serveur DHCP pour suivre les baux actifs.
4. **/etc/network/interfaces** : Bien que ce fichier ne soit pas spécifique au service DHCP, il est souvent utilisé pour configurer les interfaces réseau sur lesquelles le serveur DHCP écoutera les demandes. Vous pouvez spécifier des options de configuration supplémentaires pour les interfaces réseau dans ce fichier, telles que les adresses IP statiques, les options de routage, etc.
5. **/etc/resolv.conf** : Ce fichier contient les serveurs DNS utilisés par le système. Bien qu'il ne soit pas directement lié à la configuration du service DHCP, les serveurs DNS attribués par le serveur DHCP peuvent être mis à jour dans ce fichier sur les clients DHCP.

En utilisant ces fichiers, vous pouvez configurer et gérer efficacement le service DHCP sur votre réseau. Assurez-vous de consulter la documentation appropriée pour votre distribution Linux pour des informations spécifiques à votre environnement.

Configuration du fichier DHCP (dhcpd.conf) et du système de logs DHCP :

1. Fichier de configuration DHCP (dhcpd.conf) :

Voici un exemple de configuration pour attribuer des adresses IP en fonction des adresses hardware (MAC) des clients dans le fichier `dhcpd.conf` :

```
subnet 10.2.9.0 netmask 255.255.255.0 {
    range 10.2.9.100 10.2.9.200;
    option routers 10.2.9.1;
    option domain-name-servers 8.8.8.8;
    host client1 {
        hardware ethernet 00:11:22:33:44:55;
        fixed-address 10.2.9.101;
    }
    host client2 {
        hardware ethernet 66:77:88:99:aa:bb;
        fixed-address 10.2.9.102;
    }
    # Ajoutez d'autres hôtes ici si nécessaire
}
```

Ce bloc configure un sous-réseau (**subnet**) avec une plage d'adresses IP attribuée par **range**. Chaque client est associé à une adresse hardware (**hardware ethernet**) et à une adresse IP fixe (**fixed-address**).

2. Configuration du système de logs DHCP :

Pour configurer le système de logs pour DHCP, vous devez spécifier les paramètres de logging dans le fichier de configuration du service DHCP (`/etc/dhcp/dhcpd.conf`). Voici un exemple de configuration pour rediriger les logs DHCP vers un fichier spécifique :

```
log-facility local7;
```

Cette directive spécifie le niveau de logging (`local7`) pour le serveur DHCP. Vous pouvez ensuite configurer le système de logging (`rsyslog`, `syslog-ng`, etc.) pour rediriger les logs DHCP vers un fichier spécifique en ajoutant une règle de logging dans le fichier de configuration correspondant (`/etc/rsyslog.conf`, `/etc/syslog-ng/syslog-ng.conf`, etc.).

Par exemple, pour rediriger les logs DHCP vers un fichier spécifique avec `rsyslog`, ajoutez la ligne suivante dans `/etc/rsyslog.conf` :

```
local7.* /var/log/dhcpd.log
```

Assurez-vous que le fichier de destination `/var/log/dhcpd.log` a les autorisations appropriées pour être écrit par le service DHCP.

Après avoir apporté les modifications, redémarrez le service DHCP (`sudo systemctl restart isc-dhcp-server`) et le système de logging (`sudo systemctl restart rsyslog`).

9 Mise en place d'une sauvegarde avec rsync et cron :

1. Génération des clés SSH :

Sur le serveur, générez une paire de clés SSH (publique et privée) à l'aide de la commande suivante :

```
ssh-keygen -t rsa
```

Suivez les instructions pour générer les clés. Assurez-vous de ne pas définir de passphrase pour que la connexion SSH puisse être automatisée.

2. Configuration de l'accès SSH sans mot de passe :

Copiez la clé publique (`id_rsa.pub` par défaut) générée précédemment sur le serveur vers le client. Vous pouvez le faire en utilisant la commande `ssh-copy-id` :

```
ssh-copy-id -i ~/.ssh/id_rsa.pub user@client_ip
```

Remplacez `user` par votre nom d'utilisateur sur le client et `client_ip` par l'adresse IP du client.

3. Configuration de la sauvegarde avec rsync :

Sur le serveur, créez un script bash pour effectuer la sauvegarde avec rsync. Par exemple, créez un fichier `backup_script.sh` avec le contenu suivant :

```
#!/bin/bash
rsync -avz -e "ssh -i /path/to/private_key" /etc \
user@client_ip:/path/to/backup_folder
```

Remplacez `/path/to/private_key` par le chemin de votre clé privée SSH et `user@client_ip:/path/to/backup_folder` par le nom d'utilisateur, l'adresse IP et le chemin vers le dossier de sauvegarde sur le client.

4. Configuration de la tâche cron :

Utilisez la commande `crontab -e` pour éditer la table de tâches cron. Ajoutez une ligne pour planifier l'exécution du script de sauvegarde chaque jour à 15h :

```
0 15 * * * /path/to/backup_script.sh
```

Remplacez `/path/to/backup_script.sh` par le chemin absolu vers le script de sauvegarde que vous avez créé.

Assurez-vous que les permissions sont correctement configurées pour le script de sauvegarde afin qu'il puisse être exécuté par cron. Testez la configuration en exécutant le script manuellement et en vérifiant que la sauvegarde est effectuée comme prévu.

10 Synthèse des commandes du gestionnaire de packages Debian :

Les packages nécessaires au fonctionnement du serveur ont été installés à l'aide des commandes `apt-get`. Cependant, ce ne sont pas les seules commandes permettant d'administrer un système de gestion de packages sous Debian. En effet, il existe plusieurs outils et commandes qui peuvent être utilisés, tels que `dpkg`, `aptitude`, et `dselect`.

Voici une synthèse des commandes à connaître :

1. apt-get et aptitude :

Les commandes `apt-get` et `aptitude` sont utilisées pour gérer l'installation, la mise à jour et la suppression des packages en tenant compte des dépendances.

- `update` : Mettre à jour la liste des packages disponibles.
- `upgrade` : Mettre à jour les packages installés.
- `install` : Installer un nouveau package.
- `apt-cache search` : Rechercher dans la base de données de packages.
- `remove` : Désinstaller un package (`--purge` pour supprimer également les fichiers de configuration).
- `clean` : Effacer du disque dur les packages téléchargés.

Exemple : `apt-get update && apt-get upgrade`

2. dpkg :

Les commandes `dpkg` permettent l'installation et la désinstallation des packages mais sans gérer les dépendances.

- `-i` : Installer un package téléchargé au préalable.
- `-r` : Supprimer un package (`--purge` pour supprimer également les fichiers de configuration).
- `-s` : Connaître le nom du package ayant installé un fichier.
- `-L` : Afficher la liste des fichiers installés par un package.

Exemple : `dpkg -i package.deb`

3. dselect :

Le programme `dselect` est une alternative à `apt-get`, plus complexe à utiliser mais comportant plusieurs fonctionnalités supplémentaires.

4. Gdebi :

Utilisé pour installer un paquet Debian téléchargé avec ses dépendances.

Il est important de noter que les commandes **apt-get** et **aptitude** sont les plus couramment utilisées pour la gestion des packages sous Debian en raison de leur facilité d'utilisation et de leur capacité à gérer les dépendances de manière automatique. Cependant, **dpkg** peut être utile pour des opérations plus spécifiques ou avancées.

11 Conclusion TP1

La configuration réseau dans le fichier `/etc/network/interfaces` fournit une base solide pour assurer une connectivité stable et fiable du système Debian. Chaque directive est soigneusement configurée pour répondre aux besoins spécifiques du réseau, garantissant ainsi un fonctionnement optimal dans divers environnements réseau. Par exemple, la directive `address` définit l'adresse IP statique de l'interface `ens1f0`, permettant au système de communiquer de manière identifiable sur le réseau local. De même, la directive `gateway` spécifie l'adresse IP de la passerelle par défaut, essentielle pour router le trafic réseau vers des destinations externes. De plus, la configuration des serveurs DNS et du domaine DNS facilite la résolution des noms de domaine en adresses IP, améliorant ainsi l'accès aux services réseau et la communication au sein du réseau local. En conclusion, cette configuration réseau bien pensée assure une connectivité efficace et une communication fluide entre les périphériques, renforçant ainsi la fiabilité et la performance du réseau Debian.

12 Introduction TP 1bis

Le déploiement d'un serveur DNS constitue une étape cruciale dans la gestion d'un réseau informatique. Le Domain Name System (DNS) agit comme un pilier invisible d'Internet, traduisant les noms de domaine en adresses IP et vice versa. Ce TP propose une exploration approfondie de la mise en place d'un serveur DNS sur une plateforme Debian, offrant ainsi une immersion pratique et éducative dans le fonctionnement complexe du DNS.

Le DNS est souvent décrit comme l'épine dorsale d'Internet, facilitant la navigation en ligne en permettant aux utilisateurs de se connecter à des sites Web via des noms conviviaux plutôt qu'en mémorisant des adresses IP numériques. Toutefois, derrière cette apparente simplicité se cachent des processus sophistiqués et un réseau mondial de serveurs interconnectés.

Ce TP s'articule autour de plusieurs axes principaux. Tout d'abord, une exploration approfondie des commandes d'interrogation DNS telles que `host`, `dig` et `nslookup` permettra de comprendre comment récupérer des informations vitales sur les noms de domaine et les serveurs DNS. Ensuite, nous plongerons dans l'installation et la configuration d'un serveur DNS en utilisant ISC BIND, l'un des logiciels les plus utilisés dans ce domaine.

L'objectif est de fournir aux participants une compréhension pratique des concepts clés du DNS, tels que les zones DNS, les enregistrements de ressources (RR), et les différents modes de configuration. De la création des fichiers de zone à la gestion des enregistrements, en passant par la résolution des problèmes et la vérification du bon fonctionnement du serveur DNS, ce TP offre une expérience complète et détaillée de l'administration DNS sous Debian.

En mettant l'accent sur la pratique et l'expérimentation directe, ce TP permettra aux participants de développer des compétences essentielles en administration système tout en acquérant une compréhension approfondie du rôle central que joue le DNS dans le fonctionnement d'Internet.

13 Manuel Des Commandes (host, nslookup, dig)

Les commandes `host`, `dig` et `nslookup` constituent des outils fondamentaux pour interroger les serveurs DNS et obtenir des informations précieuses sur les noms de domaine. Leur utilisation est essentielle pour diagnostiquer les problèmes liés à la résolution DNS et comprendre la structure des domaines.

1. Commande `host` : La commande `host` permet d'effectuer des requêtes DNS simples et rapides. Son manuel détaille les différentes options disponibles, notamment la spécification du type d'enregistrement à rechercher (`-t`), l'affichage de toutes les informations disponibles (`-a`) et

l'activation du mode verbeux (-v). En utilisant `host`, les utilisateurs peuvent obtenir rapidement des informations telles que l'adresse IP associée à un nom de domaine ou le nom d'hôte associé à une adresse IP.

```
HOST(1) BIND 9

NAME
    host - DNS lookup utility

SYNOPSIS
    host [-aACdlnrstuWv] [-c class] [-N ndots] [-p port] [-R number] [-t type] [-W wait] [-m flags]
    [-v] [-V] {name} [server]

DESCRIPTION
    host is a simple utility for performing DNS lookups. It is normally used to convert names to IP
    addresses, or vice versa. When no arguments or options are given, host prints a short summary of its
    capabilities and options.

    name is the domain name that is to be looked up. It can also be a dotted-decimal IPv4 address or
    a limited IPv6 address, in which case host by default performs a reverse lookup for that address.
    [server] is an optional argument which is either the name or IP address of the name server that host should
    use, or the server or servers listed in /etc/resolv.conf.

OPTIONS
    -4      This option specifies that only IPv4 should be used for query transport. See also the
    -6      This option specifies that only IPv6 should be used for query transport. See also the
    -a      The -a ("all") option is normally equivalent to -v -t ANY. It also affects the behavior of the
             zone option.
    -A      The -A ("almost all") option is equivalent to -a, except that RRSIG, NSEC, and NSEC3 records are
             not included in the output.

Manual page host(1) line 1 (press h for help or q to quit)
```

Figure 8: Caption pour la commande `host`

2. Commande `dig` : `Dig`, abréviation de "domain information groper", offre un contrôle plus fin sur les requêtes DNS. Son manuel détaille sa syntaxe, qui permet de spécifier le nom de domaine, le type d'enregistrement recherché et éventuellement le serveur DNS à interroger. Les options disponibles permettent de personnaliser la requête DNS en fonction des besoins spécifiques de l'utilisateur, comme la spécification du type d'enregistrement (-t), la recherche inversée (-x) et la spécification du serveur DNS (@). Le manuel de `dig` fournit également des informations sur les différents types d'enregistrements DNS et leurs utilisations, ce qui permet aux utilisateurs de mieux comprendre les réponses obtenues.
3. Commande `nslookup` : Bien que considérée comme dépréciée, la commande `nslookup` est toujours largement utilisée. Son manuel fournit des informations sur son utilisation, notamment la spécification du type de requête DNS à effectuer, le type d'enregistrement DNS à rechercher et le serveur DNS à interroger. Bien que moins flexible que `dig`, `nslookup` reste un outil utile pour obtenir des informations de base sur les noms de domaine.

En explorant en détail les manuels de ces commandes, les utilisateurs peuvent acquérir une compréhension approfondie de leur fonctionnement et de leurs capacités, ce qui leur permettra de les utiliser efficace-

```

DIG(1)                                BIND 9                                DIG(1)

NAME
    dig - DNS lookup utility

SYNOPSIS
    dig [@server] [-b address] [-c class] [-f filename] [-k filename] [-m] [-p port#] [-q name] [-t type] [-v] [-x
    addr] [-y [hmac:]name:key] [ [-4] | [-6] ] [name] [type] [class] [queryopt...]

    dig [-h]

    dig [global-queryopt...] [query...]

DESCRIPTION
    dig is a flexible tool for interrogating DNS name servers. It performs DNS lookups and displays the answers
    that are returned from the name server(s) that were queried. Most DNS administrators use dig to troubleshoot
    DNS problems because of its flexibility, ease of use, and clarity of output. Other lookup tools tend to have
    less functionality than dig.

    Although dig is normally used with command-line arguments, it also has a batch mode of operation for reading
    lookup requests from a file. A brief summary of its command-line arguments and options is printed when the -h
    option is given. The BIND 9 implementation of dig allows multiple lookups to be issued from the command line.

    Unless it is told to query a specific name server, dig tries each of the servers listed in /etc/resolv.conf.
    If no usable server addresses are found, dig sends the query to the local host.

    When no command-line arguments or options are given, dig performs an NS query for "." (the root).

    It is possible to set per-user defaults for dig via $(HOME)/.digrc. This file is read and any options in it
    Manual page dig(1) line 1 (press h for help or q to quit)

```

Figure 9: Caption pour la commande dig

```

NSLOOKUP(1)                            BIND 9                            NSLOOKUP(1)

NAME
    nslookup - query Internet name servers interactively

SYNOPSIS
    nslookup [-option] [name | -] [server]

DESCRIPTION
    nslookup is a program to query Internet domain name servers. nslookup has two modes: interactive and non-in-
    teractive. Interactive mode allows the user to query name servers for information about various hosts and do-
    mains or to print a list of hosts in a domain. Non-interactive mode prints just the name and requested infor-
    mation for a host or domain.

ARGUMENTS
    Interactive mode is entered in the following cases:

    a. when no arguments are given (the default name server is used);

    b. when the first argument is a hyphen (-) and the second argument is the host name or Internet address of a
       name server.

    Non-interactive mode is used when the name or Internet address of the host to be looked up is given as the
    first argument. The optional second argument specifies the host name or address of a name server.

    Options can also be specified on the command line if they precede the arguments and are prefixed with a hy-
    phen. For example, to change the default query type to host information, with an initial timeout of 10 sec-
    onds, type:

    Manual page nslookup(1) line 1 (press h for help or q to quit)

```

Figure 10: Caption pour la commande nslookup

ment pour résoudre les problèmes liés à la résolution DNS et obtenir les informations nécessaires sur les noms de domaine.

14 Analyse de la résolution DNS pour le domaine `www.u-bourgogne.fr`

Pour déterminer l'adresse IP du serveur web associé au domaine `www.u-bourgogne.fr`, nous avons utilisé les commandes d'interrogation DNS telles que `host`, `dig` et `nslookup`. En exécutant ces commandes dans un terminal, nous avons obtenu les informations de résolution DNS pour le domaine spécifié. En analysant les résultats de chaque commande, nous avons pu identifier l'adresse IP du serveur web. Ces commandes ont été utiles pour obtenir rapidement et efficacement les informations nécessaires sur l'infrastructure DNS du domaine `www.u-bourgogne.fr`, ce qui nous a permis de poursuivre notre configuration et notre analyse du serveur DNS.


```

root@agence:~# host www.u-bourgogne.fr
www.u-bourgogne.fr is an alias for tokyo.dmz.u-bourgogne.fr.
tokyo.dmz.u-bourgogne.fr has address 193.52.234.14

```

Figure 11: Caption pour la commande host

```

root@agence:~# nslookup www.u-bourgogne.fr
Server:      10.2.10.1
Address:     10.2.10.1#53

Non-authoritative answer:
www.u-bourgogne.fr      canonical name = tokyo.dmz.u-bourgogne.fr.
Name:   tokyo.dmz.u-bourgogne.fr
Address: 193.52.234.14

```

Figure 12: Caption pour la commande nslookup

```

root@agence:~# dig www.u-bourgogne.fr

; <<>> DiG 9.18.24-1-Debian <<>> www.u-bourgogne.fr
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49520
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 938b071616ec867a0100000066194b06a1be52084e1830da (good)
;; QUESTION SECTION:
;www.u-bourgogne.fr.      IN      A

;; ANSWER SECTION:
www.u-bourgogne.fr.      3563    IN      CNAME   tokyo.dmz.u-bourgogne.fr.
tokyo.dmz.u-bourgogne.fr. 3563    IN      A       193.52.234.14

;; Query time: 0 msec
;; SERVER: 10.2.10.1#53(10.2.10.1) (UDP)
;; WHEN: Fri Apr 12 16:53:58 CEST 2024
;; MSG SIZE rcvd: 129

```

Figure 13: Caption pour la commande dig

15 Analyse de la Réponse DNS et Changement de Serveur DNS

Lors de l'interrogation d'un serveur DNS spécifique pour obtenir des informations sur un domaine donné, il est crucial de comprendre quel serveur DNS répond à la requête et comment changer de serveur DNS si nécessaire. Cette section examine en détail la réponse DNS obtenue lors de l'utilisation de la commande `dig` et explore les mécanismes pour choisir un serveur DNS spécifique.

15.1 Identification du Serveur DNS Répondant

Lors de l'exécution de la commande `dig @8.8.8.8 agence.lyon`, le serveur DNS qui a répondu à la requête est identifié dans la section "SERVER" des résultats de la commande. Cette section indique clairement l'adresse IP du serveur DNS utilisé pour répondre à la requête.

15.2 Changement de Serveur DNS avec dig

Pour changer de serveur DNS avec la commande `dig`, vous pouvez spécifier l'adresse IP du serveur DNS souhaité à l'aide de l'option `@`. Par exemple, en ajoutant `@8.8.8.8` à la commande `dig`, vous indiquez explicitement que vous souhaitez interroger le serveur DNS avec l'adresse IP 8.8.8.8 pour obtenir des informations sur le domaine "agence.lyon". Cette flexibilité dans le choix du serveur DNS permet d'effectuer des tests de résolution DNS spécifiques et d'analyser les réponses de différents serveurs DNS.

En résumé, cette section fournit un aperçu de la manière dont les réponses DNS sont analysées et comment choisir un serveur DNS spécifique pour effectuer des requêtes DNS. Elle met en lumière l'importance de comprendre la source des résultats DNS et de pouvoir sélectionner des serveurs DNS appropriés en fonction des besoins spécifiques de résolution DNS.

16 Analyse des Serveurs de Noms pour le Domaine "u-bourgogne.fr"

Dans cette section, nous examinerons en détail les résultats des commandes `host -t ns u-bourgogne.fr` et `dig u-bourgogne.fr ns`, qui ont été utilisées pour obtenir des informations sur les serveurs de noms (NS) associés au domaine "u-bourgogne.fr". Cette analyse nous permettra de comprendre la structure et la configuration de l'infrastructure DNS du domaine cible, offrant ainsi un aperçu précieux pour sa gestion et son administration.

16.1 Résultats de la Commande `host -t ns u-bourgogne.fr`

La commande `host -t ns u-bourgogne.fr` a retourné une liste des serveurs de noms autoritaires pour le domaine "u-bourgogne.fr". Chaque serveur de nom a été identifié par son nom de domaine associé, fournissant ainsi une vue claire des ressources DNS responsables de la résolution des requêtes pour ce domaine.

16.2 Résultats de la Commande `dig u-bourgogne.fr ns`

De même, la commande `dig u-bourgogne.fr ns` a permis d'effectuer une requête de type NS pour le domaine "u-bourgogne.fr" et a retourné une liste détaillée des serveurs de noms autoritaires associés à ce domaine. Les résultats de cette commande ont fourni des informations supplémentaires telles que la durée de vie des enregistrements et les enregistrements de type NS pour chaque serveur de noms.

16.3 Analyse Comparative

En comparant les résultats des deux commandes, nous avons pu observer des similitudes dans les informations fournies, avec chaque commande confirmant les serveurs de noms autoritaires identifiés pour le domaine "u-bourgogne.fr". Cette analyse approfondie des serveurs de noms nous permet de mieux comprendre l'architecture DNS du domaine cible et fournit une base solide pour sa gestion et son administration efficaces.

Cette section offre ainsi une vue détaillée et informatique sur les serveurs de noms pour le domaine "u-bourgogne.fr", contribuant ainsi à une meilleure compréhension de son infrastructure DNS.

```
root@agence:/# host -t ns u-bourgogne.fr
u-bourgogne.fr name server dns1.u-bourgogne.fr.
u-bourgogne.fr name server ufc.univ-fcomte.fr.
u-bourgogne.fr name server dns2.u-bourgogne.fr.
```

Figure 14: Caption pour la liste des serveurs

17 Consultation DNS et Suivi de la Trace avec la Commande dig

Dans cette section, nous examinerons l'utilisation de la commande `dig` pour interroger les serveurs DNS et suivre la trace des requêtes DNS pour différents domaines, y compris "u-bourgogne.fr", "fr", et la racine ".". Nous observerons également le résultat de la commande `dig` avec l'option `+trace` pour le domaine "www.u-bourgogne.fr".

17.1 Interrogation des Domaines avec la Commande dig

Nous avons utilisé la commande `dig` pour interroger les serveurs DNS pour les domaines "u-bourgogne.fr" et "fr", ainsi que pour la racine ".". Ces requêtes nous ont permis d'obtenir des informations sur les enregistrements DNS associés à ces domaines, notamment les serveurs de noms autoritaires et les enregistrements de type A (adresse IP) ou NS (serveur de noms).

17.2 Observation du Résultat avec l'Option +trace

En utilisant l'option `+trace` avec la commande `dig`, nous avons suivi le cheminement des requêtes DNS depuis la racine jusqu'au serveur autoritaire pour le domaine "www.u-bourgogne.fr". Cette fonctionnalité nous a permis de visualiser chaque étape de la résolution DNS, montrant les serveurs DNS interrogés à chaque niveau de la hiérarchie DNS.

17.3 Analyse des Résultats

Les résultats des requêtes `dig` ont fourni des informations précieuses sur la configuration et la résolution DNS pour les domaines interrogés. Nous avons pu observer les serveurs de noms autoritaires, les adresses IP associées aux domaines, ainsi que le cheminement des requêtes DNS à travers la hiérarchie DNS jusqu'au serveur autoritaire final pour le domaine "www.u-bourgogne.fr".

Cette section démontre l'utilisation pratique de la commande `dig` pour explorer et comprendre le fonctionnement du système DNS, ce qui est essentiel pour la gestion et la configuration efficaces des infrastructures réseau.

18 Gestion des Serveurs Racines DNS

Les serveurs DNS racines sont une composante essentielle de l'Internet. Ils sont au nombre de 13 et sont chargés de gérer le «.». Ces serveurs sont capables de répondre directement aux requêtes pour des enregistrements stockés ou mis en cache dans la zone racine. Ils peuvent également rediriger d'autres requêtes vers le serveur du domaine de premier niveau (TLD) approprié.

Il est important de noter que chaque adresse IP parmi les 13 a plusieurs serveurs derrière elle. Ces serveurs utilisent le routage Anycast pour distribuer les requêtes en fonction de la charge et de la proximité. Actuellement, il y a plus de 600 serveurs racine DNS différents répartis sur tous les continents habités de la planète.

L'ICANN (Internet Corporation for Assigned Names and Numbers) est l'organisation qui gère les serveurs pour l'une des 13 adresses IP de la zone racine. Elle délègue la gestion des 12 autres adresses IP à diverses organisations. Pour obtenir des informations précises sur le serveur qui gère le «e», il est recommandé de consulter le site root-servers.org.

En somme, les serveurs DNS racines jouent un rôle crucial dans le fonctionnement de l'Internet. Ils assurent la résolution des noms de domaine en adresses IP, permettant ainsi aux utilisateurs d'accéder aux sites web et aux services en ligne.

19 Gestion de la Configuration du Serveur DNS

19.1 Gestion de la Configuration du Serveur DNS sur un Système Linux

La configuration du serveur DNS sur un système Linux est centralisée dans le fichier système `/etc/resolv.conf`. Ce fichier spécifie les adresses IP des serveurs DNS utilisés par le système pour résoudre les noms de domaine en adresses IP.

19.2 Consultation de la Configuration Actuelle

Pour vérifier la configuration actuelle du serveur DNS, l'outil `nano`, un éditeur de texte couramment utilisé, peut être employé pour ouvrir le fichier `/etc/resolv.conf`. Cette commande ouvre le fichier dans `nano`. Dans cet exemple, `10.2.9.1` représente l'adresse IP du serveur DNS actuellement configuré sur le système.

19.3 Modification de la Configuration DNS

Pour ajuster la configuration du serveur DNS, les modifications peuvent être apportées directement dans l'éditeur `nano`. Vous pouvez ajouter, modifier ou supprimer des lignes commençant par `nameserver`, suivies de l'adresse IP du serveur DNS souhaité.

Il est primordial de noter que toute modification dans ce fichier peut impacter la capacité du système à résoudre les noms de domaine. Il est donc recommandé de sauvegarder le fichier original avant toute modification et de s'assurer de la validité et de la fonctionnalité des adresses IP des serveurs DNS ajoutés.



```
GNU nano 7.2 resolv.conf
domain agence.lyon
nameserver 10.2.9.1
```

Figure 15: Caption pour le fichier `resolv.conf`

20 Implémentation et Configuration Personnalisée du Fichier `/etc/hosts`

Dans cette section, nous abordons l'implémentation et la configuration personnalisée du fichier `/etc/hosts`, adaptées à nos besoins spécifiques en tant qu'utilisateur. Cette approche nous permet de gérer efficacement la résolution des noms de domaine sur notre système selon nos préférences et nos exigences.

20.1 Personnalisation du Fichier `/etc/hosts`

En tant qu'utilisateur, la personnalisation du fichier `/etc/hosts` nous offre la possibilité de configurer le système selon nos besoins individuels. Cette flexibilité nous permet d'ajouter des entrées pour des noms de domaine locaux utilisés fréquemment dans nos projets de développement, des alias pour des adresses IP internes, ou encore de bloquer l'accès à des sites web spécifiques en redirigeant leur adresse IP vers une adresse locale.

20.2 Utilisation en Développement Local

Dans un contexte de développement local, le fichier `/etc/hosts` peut être utilisé pour créer des alias conviviaux pour les adresses IP locales, simplifiant ainsi l'accès aux serveurs de développement, aux bases de données et à d'autres ressources essentielles pour le développement et les tests.

Par exemple, en ajoutant une entrée comme `127.0.0.1 localhost.example.com` à notre fichier `/etc/hosts`, nous pouvons accéder à notre serveur local en utilisant le nom de domaine convivial `localhost.example.com`, au lieu de devoir mémoriser ou saisir l'adresse IP locale chaque fois que nous souhaitons y accéder.

20.3 Blocage de Sites Web Indésirables

En plus de faciliter l'accès aux ressources locales, le fichier `/etc/hosts` peut également être utilisé pour bloquer l'accès à des sites web indésirables en redirigeant leur adresse IP vers une adresse locale telle que `127.0.0.1`. Cette approche est souvent utilisée pour restreindre l'accès à des sites web nuisibles ou non souhaités, offrant ainsi un contrôle supplémentaire sur la navigation en ligne.

Par exemple, en ajoutant une entrée comme `127.0.0.1 www.example.com` à notre fichier `/etc/hosts`, nous redirigeons toute requête vers le site web `www.example.com` vers notre propre machine locale, empêchant ainsi l'accès à ce site depuis notre navigateur web.

20.4 Gestion Prudente du Fichier `/etc/hosts`

Il est important de gérer le fichier `/etc/hosts` avec prudence, en veillant à ne pas introduire d'erreurs de syntaxe ou de conflits d'adresses IP qui pourraient perturber la résolution des noms de domaine sur notre système. En sauvegardant régulièrement le fichier original et en effectuant des tests après chaque modification, nous pouvons éviter les problèmes potentiels et assurer un fonctionnement fluide de la résolution des noms de domaine sur notre système.

En conclusion, la personnalisation du fichier `/etc/hosts` offre une flexibilité précieuse pour gérer la résolution des noms de domaine sur notre système, nous permettant de configurer et d'optimiser la résolution des noms de domaine selon nos besoins spécifiques en tant qu'utilisateur.

21 Conclusion TP1bis

La configuration et l'administration d'un serveur DNS sur une plateforme Debian sont des compétences essentielles pour tout administrateur système ou réseau. Ce TP a offert une immersion pratique dans le monde complexe du DNS, explorant les commandes d'interrogation DNS, l'installation et la configuration d'un serveur DNS avec ISC BIND, l'analyse des réponses DNS, et la gestion de la résolution des noms de domaine sur un système Linux.

En mettant l'accent sur la pratique et l'expérimentation directe, ce TP a permis aux participants de développer des compétences pratiques en administration système tout en acquérant une compréhension approfondie du rôle crucial que joue le DNS dans le fonctionnement d'Internet. En explorant les commandes d'interrogation DNS, la configuration du serveur DNS, et la personnalisation du fichier `/etc/hosts`, les participants ont pu acquérir une expérience pratique et éducative dans la gestion et l'administration des infrastructures DNS sous Debian.

En conclusion, ce TP constitue une étape importante dans le parcours de formation des administrateurs système et réseau, offrant une expérience pratique et immersive dans le monde complexe du DNS et des infrastructures réseau. En développant des compétences pratiques en administration système et en acquérant une compréhension approfondie du DNS, les participants sont mieux préparés à gérer et à administrer efficacement les infrastructures réseau dans un environnement professionnel.

22 Introduction TP2

Dans le cadre de notre projet, nous avons entrepris la mise en place et la configuration d'un serveur web sur une distribution Debian. Cette partie détaille les étapes que nous avons suivies pour installer, configurer et sécuriser ce serveur, ainsi que les différentes tâches administratives effectuées pour assurer son bon fonctionnement.

Nous débutons par la mise en place du serveur HTTP, en décrivant de manière précise les commandes utilisées pour son installation et sa configuration initiale. Ensuite, nous abordons la création de comptes utilisateurs dédiés au développement web, en attribuant des permissions spécifiques à chaque utilisateur selon ses besoins.

L'installation et la configuration des bases de données MariaDB, PostgreSQL et MySQL sont ensuite détaillées, mettant en lumière les différentes étapes pour créer des bases de données, des utilisateurs et leur attribuer les privilèges nécessaires. Nous explorons également l'installation de PHP et son intégration avec les bases de données, notamment en utilisant PDO pour assurer une abstraction efficace.

Par la suite, nous examinons en détail la mise en place des règles de filtrage IPTables pour sécuriser notre serveur, en suivant des recommandations spécifiques pour limiter l'accès aux services essentiels et garantir une communication sécurisée entre le client et le serveur.

Nous concluons ce rapport en mettant en évidence l'automatisation des tâches administratives, telles que la sauvegarde régulière des bases de données et la configuration automatique du routage et de la translation d'adresses au démarrage du serveur.

Ce document vise à fournir un guide complet et détaillé pour la mise en place et la gestion d'un serveur web sous Debian, tout en soulignant les bonnes pratiques de sécurité et d'administration système.

23 Installation d'un serveur HTTP

Lorsque nous débutons la mise en place de notre environnement web sur Debian, la première étape fondamentale est l'installation du serveur HTTP Apache, un pilier de l'infrastructure web. Avant toute action, nous nous assurons de mettre à jour notre système Debian pour garantir la stabilité et la sécurité de l'ensemble. Pour ce faire, nous exécutons la commande `sudo apt update`, qui met à jour la liste des paquets disponibles, suivie de `sudo apt upgrade`, qui installe les mises à jour disponibles. Une fois cette phase préliminaire terminée, nous nous tournons vers l'installation d'Apache.

L'installation d'Apache se fait en utilisant la commande `sudo apt install apache2`. Cette commande déclenche le téléchargement et l'installation d'Apache ainsi que de toutes ses dépendances à partir des référentiels Debian. Une fois l'installation achevée, Apache est automatiquement démarré et ajouté aux services système, mais il est toujours sage de vérifier son statut pour s'assurer qu'il fonctionne correctement. Cela se fait en tapant `sudo systemctl status apache2` dans le terminal, où nous devrions voir un message indiquant que le service est actif et en cours d'exécution.

Pour confirmer que notre installation d'Apache est fonctionnelle, nous ouvrons un navigateur web et accédons à l'adresse IP de notre serveur Debian. Par défaut, Apache affiche une page de test qui indique que tout fonctionne correctement. Cette page par défaut est généralement située dans le répertoire `/var/www/html/`.

Pour personnaliser davantage la configuration d'Apache, nous nous tournons vers les fichiers de configuration situés dans le répertoire `/etc/apache2/`. Par exemple, le fichier `/etc/apache2/apache2.conf` contient les configurations globales d'Apache, telles que les paramètres de sécurité et les directives du serveur. De plus, pour héberger plusieurs sites sur un même serveur, nous utilisons des virtual hosts. Les fichiers de configuration des virtual hosts sont stockés dans le répertoire `/etc/apache2/sites-available/`. Pour activer un virtual host, nous créons un lien symbolique du fichier de configuration du virtual host vers le répertoire `/etc/apache2/sites-enabled/` à l'aide de la commande `sudo ln -s /etc/apache2/sites-available/mon_site.conf /etc/apache2/sites-enabled/`. Enfin, nous appliquons les changements en redémarrant Apache avec `sudo systemctl reload apache2`.

En conclusion, l'installation et la configuration d'Apache nécessitent une série d'étapes méthodiques, mais relativement simples, qui garantissent un serveur HTTP robuste et fonctionnel. En suivant attentivement ces étapes et en comprenant les paramètres de configuration disponibles, nous pouvons créer un environnement web personnalisé et sécurisé pour répondre à nos besoins spécifiques.

24 Création d'un compte utilisateur web1

Pour assurer un environnement de développement web efficace et sécurisé, nous débutons par la création d'un compte Unix dédié à l'utilisateur en charge du développement d'applications web. Nous utilisons la commande `adduser` pour créer ce nouvel utilisateur. Par exemple, pour créer un utilisateur nommé

"web1", nous exécutons la commande suivante : `sudo adduser web1`. Cette commande nous guide à travers le processus de création du compte utilisateur, en incluant la définition d'un mot de passe et la fourniture d'informations supplémentaires.

Une fois le compte utilisateur créé, nous procédons à la configuration du serveur web pour associer le répertoire `www` de l'utilisateur à un emplacement spécifique. Pour cela, nous déplaçons le répertoire `www` vers le dossier `/srv/web1` en utilisant la commande `mv`. Par exemple : `sudo mv /var/www /srv/web1`. Ensuite, nous devons nous assurer que les bons droits et propriétaires sont attribués à l'arborescence principale du site. Pour cela, nous utilisons les commandes `chown` et `chmod` pour définir les permissions appropriées sur les fichiers et répertoires du site. Par exemple : `sudo chown -R web1:www-data /srv/web1` et `sudo chmod -R 755 /srv/web1`.

Une fois la configuration des permissions terminée, nous créons un alias dans la configuration du serveur web pour pointer sur le site. Pour ce faire, nous éditons le fichier de configuration du serveur web (généralement situé dans `/etc/apache2/sites-available/`) et ajoutons un alias pointant sur le répertoire `www` de l'utilisateur. Par exemple :

```
Alias /web1 /srv/web1/www
<Directory /srv/web1/www>
Options Indexes FollowSymLinks
AllowOverride All
Require all granted
</Directory>
```

Enfin, pour rendre le site accessible depuis un navigateur, nous devons configurer le DNS pour inclure une entrée pointant vers le site. Supposons que nous voulons que le site soit accessible depuis `http://www.agence.lyon/web1`. Nous devons configurer l'entrée DNS pour rediriger ce domaine vers l'adresse IP de notre serveur. Une fois cette configuration terminée et les modifications DNS propagées, le site sera accessible depuis l'URL spécifiée.

En conclusion, en suivant ces étapes détaillées, nous avons réussi à créer un compte Unix pour un utilisateur développant des applications web, à configurer le serveur web pour associer le répertoire `www` de l'utilisateur à un emplacement spécifique, et à rendre le site accessible via une URL spécifique. Ce processus garantit un environnement de développement web sécurisé et bien configuré pour le développement d'applications web.

25 Création d'un compte utilisateur web2

Pour assurer un environnement de développement web complet et diversifié, nous procédons à la création d'un compte Unix dédié à un deuxième utilisateur, ici nommé "web2". Nous utilisons la commande `adduser` pour créer ce nouvel utilisateur. Par exemple, pour créer un utilisateur nommé "web2", nous exécutons la commande suivante : `sudo adduser web2`. Cette commande nous guide à travers le processus de création du compte utilisateur, incluant la définition d'un mot de passe et la fourniture d'informations supplémentaires.

Une fois le compte utilisateur "web2" créé, nous procédons à la configuration du serveur web pour associer le répertoire `www` de cet utilisateur à un emplacement spécifique. Nous déplaçons le répertoire `www` vers le dossier `/srv/web2` en utilisant la commande `mv`. Par exemple : `sudo mv /var/www /srv/web2`. Ensuite, nous nous assurons que les bons droits et propriétaires sont attribués à l'arborescence principale du site. Pour cela, nous utilisons les commandes `chown` et `chmod`. Par exemple : `sudo chown -R web2:www-data /srv/web2` et `sudo chmod -R 755 /srv/web2`.

Après avoir configuré les permissions, nous créons un nouveau Virtualhost Directory dans la configuration du serveur web pour pointer vers le site web2. Nous éditons le fichier de configuration du serveur web (généralement situé dans `/etc/apache2/sites-available/`) et ajoutons une nouvelle section pour le Virtualhost Directory. Par exemple :

```
<VirtualHost *:80>
ServerName informatique.agence.dijon
DocumentRoot /srv/web2/www
<Directory /srv/web2/www>
Options Indexes FollowSymLinks
```

```
AllowOverride All
Require all granted
</Directory>
</VirtualHost>
```

Enfin, pour que le site soit accessible depuis un navigateur, nous devons configurer le DNS pour inclure une entrée pointant vers le site. Supposons que nous voulons que le site soit accessible depuis `http://informatique.agence.lyon/`. Nous devons configurer l'entrée DNS pour rediriger ce domaine vers l'adresse IP de notre serveur. Une fois cette configuration terminée et les modifications DNS propagées, le site sera accessible depuis l'URL spécifiée.

En conclusion, en suivant ces étapes détaillées, nous avons réussi à créer un deuxième compte Unix pour un utilisateur, à configurer un nouveau Virtualhost Directory sur le serveur web, et à rendre le site accessible via une nouvelle URL. Cela permet d'offrir un environnement de développement web diversifié et sécurisé pour plusieurs utilisateurs et projets.

26 Création d'un compte utilisateur web3

Pour répondre aux besoins des utilisateurs occasionnels et leur offrir un moyen simple de publier des pages personnelles, nous créons un compte Unix dédié à un troisième utilisateur, ici nommé "web3". Contrairement aux utilisateurs précédents, nous configurons le répertoire personnel de cet utilisateur sous

`/home/web3`. Nous utilisons la commande `adduser` pour créer ce nouvel utilisateur. Par exemple, pour créer un utilisateur nommé "web3", nous exécutons la commande suivante : `sudo adduser web3`. Cette commande nous guide à travers le processus de création du compte utilisateur, incluant la définition d'un mot de passe et la fourniture d'informations supplémentaires.

Une fois le compte utilisateur "web3" créé, nous mettons en place l'utilisation d'un `public_html` pour permettre à cet utilisateur de publier des pages personnelles. Par défaut, Apache est configuré pour servir les pages situées dans le répertoire `public_html` du répertoire personnel de chaque utilisateur. Ainsi, nous créons le répertoire `public_html` dans le répertoire `/home/web3` de l'utilisateur "web3" : `mkdir /home/web3/public_html`. Nous assurons également que les bons droits et propriétaires sont attribués à ce répertoire, en utilisant les commandes `chown` et `chmod`. Par exemple : `sudo chown -R web3:web3`

`/home/web3/public_html` et `sudo chmod -R 755 /home/web3/public_html`.

Enfin, pour que les pages publiées par l'utilisateur "web3" soient accessibles depuis un navigateur, nous utilisons l'URL spéciale de la forme `http://www.agence.lyon/web3`. Par défaut, Apache est configuré pour interpréter cette URL comme une requête vers le répertoire `public_html` du répertoire personnel de l'utilisateur "web3". Ainsi, une fois que l'utilisateur "web3" publie des pages dans son répertoire `public_html`, elles seront accessibles via cette URL spécifique.

En conclusion, en suivant ces étapes détaillées, nous avons réussi à créer un compte Unix pour un troisième utilisateur, à mettre en place l'utilisation d'un `public_html` pour publier des pages personnelles et à rendre ces pages accessibles via une URL spécifique. Cela permet aux utilisateurs occasionnels de partager facilement du contenu en ligne, tout en maintenant une organisation sécurisée et structurée de leur espace personnel sur le serveur web.

27 Installation de Gparted et création d'une nouvelle partition

La première étape consiste à installer GParted, un outil graphique de partitionnement, qui simplifie grandement le processus de gestion des partitions. Nous commençons par mettre à jour la liste des paquets disponibles et installer GParted en utilisant les commandes suivantes dans le terminal :

```
sudo apt update
sudo apt install gparted
```

Une fois l'installation terminée, nous lançons GParted en exécutant la commande :

```
sudo gparted
```


GParted s'ouvre alors avec une interface graphique conviviale, affichant une représentation visuelle des partitions de notre disque dur. Nous localisons l'espace libre disponible sur le disque dur, qui sera utilisé pour créer la nouvelle partition destinée à héberger les bases de données.

En cliquant avec le bouton droit de la souris sur l'espace libre, nous sélectionnons l'option "Nouvelle" pour créer une nouvelle partition. Une fenêtre s'ouvre alors nous permettant de spécifier la taille de la partition et le système de fichiers à utiliser. Pour héberger des bases de données, un système de fichiers comme ext4 est souvent recommandé pour sa robustesse.

Une fois la partition créée, nous la formaterons avec le système de fichiers choisi. Pour ce faire, nous sélectionnons la nouvelle partition, cliquons avec le bouton droit de la souris et choisissons l'option "Formater vers" en sélectionnant le système de fichiers désiré.

Ensuite, nous configurons le point de montage de la partition nouvellement créée en le définissant sur /BD, qui sera le répertoire racine pour héberger les bases de données. Cette étape se fait en cliquant avec le bouton droit de la souris sur la partition, en sélectionnant "Gérer les étiquettes" et en spécifiant "/BD" comme point de montage.

Enfin, nous appliquons les modifications en cliquant sur le bouton "Appliquer" dans GParted. Une fois les opérations terminées, la nouvelle partition sera prête à être utilisée comme espace de stockage pour héberger les bases de données.

En conclusion, en suivant attentivement ces étapes détaillées avec GParted, nous avons installé avec succès l'outil, créé une nouvelle partition sur l'espace libre du disque dur, et configuré celle-ci avec comme point de montage /BD pour héberger les bases de données. Cette méthode offre une solution pratique et efficace pour gérer l'espace de stockage dédié aux données de nos applications.

28 Installation de MariaDB

Pour débiter l'installation de MariaDB, nous utilisons la commande suivante dans le terminal :

```
sudo apt update
sudo apt install mariadb-server
```

Ces commandes mettent à jour la liste des paquets disponibles et déploient MariaDB sur notre système Debian. À la fin de l'installation, MariaDB est automatiquement démarré et ajouté aux services système. Nous pouvons vérifier l'état de MariaDB en exécutant la commande suivante :

```
sudo systemctl status mariadb
```

Le résultat devrait confirmer que le service est actif et en cours d'exécution.

Une fois MariaDB installé et opérationnel, nous pouvons accéder à son interface de ligne de commande en utilisant la commande :

```
sudo mysql
```

Dans cette interface, nous créons une nouvelle base de données en exécutant la commande suivante :

```
CREATE DATABASE bd_tp2;
```

Ensuite, nous définissons un nouvel utilisateur et lui attribuons les privilèges nécessaires pour accéder à la base de données nouvellement créée. Voici un exemple de commande :

```
CREATE USER 'utilisateur'@'localhost' IDENTIFIED BY 'mot_de_passe';
GRANT ALL PRIVILEGES ON bd_tp2.* TO 'utilisateur'@'localhost';
FLUSH PRIVILEGES;
```

Une fois la base de données et l'utilisateur configurés, nous quittons l'interface de ligne de commande de MariaDB en utilisant la commande :

```
exit;
```

En suivant ces étapes détaillées, nous avons installé MariaDB, créé une base de données et un utilisateur dans le SGBD, prêts à être utilisés pour nos applications web ou autres projets nécessitant une gestion de base de données.

29 Installation et validation de PHP

Pour commencer, nous installons PHP sur notre serveur en utilisant la commande suivante dans le terminal :

```
sudo apt install php
```

Cela va installer PHP ainsi que tous les modules et dépendances nécessaires à son fonctionnement. Une fois l'installation terminée, nous devons vérifier si PHP fonctionne correctement avec Apache. Pour cela, nous créons un fichier PHP de test dans le répertoire de documents d'Apache en utilisant la commande suivante :

```
sudo nano /var/www/html/info.php
```

Dans ce fichier, nous ajoutons le code suivant pour afficher les informations sur PHP :

```
<?php
phpinfo();
?>
```

Ensuite, nous enregistrons le fichier et quittons l'éditeur de texte. Pour accéder à ce fichier via un navigateur web, nous devons connaître l'adresse IP de notre serveur. En tapant cette adresse IP suivi de /info.php dans la barre d'adresse du navigateur, nous devrions voir une page affichant les informations sur PHP, confirmant ainsi son bon fonctionnement avec Apache.

Maintenant que PHP est installé et fonctionne correctement, nous pouvons valider l'installation du trio Apache, PHP et MySQL en créant un programme PHP pour se connecter à la base de données MySQL et afficher le contenu d'une table. Pour ce faire, nous créons un nouveau fichier PHP dans le répertoire de documents d'Apache en utilisant la commande suivante :

```
sudo nano /var/www/html/bd_connexion.php
```

Dans ce fichier, nous écrivons le code PHP suivant pour se connecter à la base de données MySQL et afficher le contenu d'une table spécifique :

```
<?php
$servername = "localhost";
$username = "utilisateur";
$password = "mot_de_passe";
$dbname = "bd_tp2";

// Connexion à la base de données
$conn = mysqli_connect($servername, $username, $password, $dbname);

// Vérifier la connexion
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

// Requête SQL pour récupérer les données de la table
$sql = "SELECT * FROM nom_de_la_table";
$result = mysqli_query($conn, $sql);

if (mysqli_num_rows($result) > 0) {
    // Affichage des données sous forme de tableau
    echo "<table border='1'>";
    echo "<tr><th>Colonne1</th><th>Colonne2</th></tr>";
    while($row = mysqli_fetch_assoc($result)) {
        echo "<tr><td>".$row["colonne1"]."</td><td>".$row["colonne2"]."
```

```

        "</td></tr>";
    }
    echo "</table>";
} else {
    echo "0 results";
}

// Fermer la connexion à la base de données
mysqli_close($conn);
?>

```

Dans ce code, nous devons remplacer "utilisateur", "mot_de_passe", "bd_tp2" et "users" par les informations appropriées pour notre configuration MySQL.

Une fois le fichier enregistré, nous pouvons accéder à ce fichier via un navigateur web en tapant l'adresse IP de notre serveur suivi de /mysql.php dans la barre d'adresse. Nous devrions alors voir le contenu de la table MySQL affiché sur la page web, sous le titre "Données MySQL".

30 Installation de PostgreSQL

Pour commencer, nous installons PostgreSQL sur notre serveur en utilisant la commande suivante dans le terminal :

```
sudo apt install postgresql postgresql-contrib
```

Cela va installer PostgreSQL ainsi que les extensions supplémentaires nécessaires. Une fois l'installation terminée, PostgreSQL est automatiquement démarré et ajouté aux services système.

Maintenant que PostgreSQL est installé, nous devons vérifier si le service est en cours d'exécution. Nous utilisons la commande suivante pour vérifier le statut de PostgreSQL :

```
sudo systemctl status postgresql
```

Le résultat devrait indiquer que le service est actif et en cours d'exécution.

Ensuite, nous pouvons accéder à l'interface de ligne de commande de PostgreSQL en utilisant la commande suivante :

```
sudo -u postgres psql
```

Dans cette interface, nous pouvons créer une nouvelle base de données en utilisant la commande suivante :

```
CREATE DATABASE bd_tp2;
```

Nous créons également un nouvel utilisateur et lui attribuons des privilèges sur la base de données nouvellement créée :

```
CREATE USER utilisateur WITH PASSWORD 'etudiant';
GRANT ALL PRIVILEGES ON DATABASE bd_tp2 TO utilisateur;
```

Une fois que la base de données et l'utilisateur sont configurés, nous quittons l'interface de ligne de commande de PostgreSQL en utilisant la commande :

```
\q
```

Maintenant que PostgreSQL est prêt à être utilisé, nous pouvons créer une table similaire à celle de MySQL. Pour cela, nous utilisons la même structure de table et les mêmes colonnes que celles utilisées pour MySQL.

Une fois la table créée et des données ajoutées à PostgreSQL, nous pouvons utiliser PHP avec le module PostgreSQL pour se connecter à la base de données et afficher le contenu de la table. Nous utilisons un script PHP similaire à celui utilisé pour MySQL, mais en utilisant les fonctions de PostgreSQL pour se connecter à la base de données et exécuter les requêtes SQL nécessaires.

Enfin, nous intégrons ce script PHP à la même page web précédente de "web1" sous le titre "Données PostgreSQL" afin d'afficher les informations de la table PostgreSQL sur la même page que celles de MySQL.

31 Utilisation de PDO en PHP pour accéder et afficher les données des bases de données MySQL et PostgreSQL

Pour commencer, nous installons le module PDO dans PHP en utilisant la commande suivante dans le terminal :

```
sudo apt install php-pdo php-mysql php-pgsql
```

Cela va installer le module PDO ainsi que les pilotes MySQL et PostgreSQL pour PDO. Une fois l'installation terminée, nous devons redémarrer le service Apache pour prendre en compte les modifications :

```
sudo systemctl restart apache2
```

Maintenant que le module PDO est installé, nous pouvons développer un programme PHP pour se connecter aux bases de données MySQL et PostgreSQL précédemment créées et afficher leur contenu.

Tout d'abord, pour afficher les données de la base MySQL, nous utilisons le code PHP suivant dans un fichier nommé `mysql_pdo.php` :

```
<?php
try {
    $pdo = new PDO("mysql:host=localhost;dbname=bd_tp2",
        "utilisateur", "mot_de_passe");
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $stmt = $pdo->query("SELECT * FROM nom_de_la_table");
    $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);

    echo "<h2>Données MySQL avec PDO</h2>";
    echo "<table border='1'>";
    echo "<tr><th>Colonne1</th><th>Colonne2</th></tr>";
    foreach ($rows as $row) {
        echo "<tr><td>{$row['colonne1']}</td><td>{$row['colonne2']}</td></tr>";
    }
    echo "</table>";
} catch (PDOException $e) {
    echo "Erreur : " . $e->getMessage();
}
?>
```

Ensuite, pour afficher les données de la base PostgreSQL, nous utilisons le code PHP suivant dans un fichier nommé `postgresql_pdo.php` :

```
<?php
try {
    $pdo = new PDO("pgsql:host=localhost;dbname=bd_tp2", "utilisateur", "etudiant");
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);

    $stmt = $pdo->query("SELECT * FROM users");
    $rows = $stmt->fetchAll(PDO::FETCH_ASSOC);

    echo "<h2>Données PostgreSQL avec PDO</h2>";
    echo "<table border='1'>";
    echo "<tr><th>Colonne1</th><th>Colonne2</th></tr>";
    foreach ($rows as $row) {
        echo "<tr><td>{$row['colonne1']}</td><td>{$row['colonne2']}</td></tr>";
    }
}
```

```

        </td></tr>";
    }
    echo "</table>";
} catch (PDOException $e) {
    echo "Erreur : " . $e->getMessage();
}
?>

```

Enfin, nous intégrons ces scripts PHP à la page web de "web1" pour afficher les données des bases MySQL et PostgreSQL sous les titres "Données MySQL" et "Données PostgreSQL" respectivement.

32 Mise en place de la sauvegarde régulière des bases de données MySQL et PostgreSQL

Pour assurer la sauvegarde de nos bases de données MySQL et PostgreSQL, nous mettons en place des scripts de sauvegarde réguliers via des tâches planifiées.

Tout d'abord, pour MySQL, nous utilisons la commande `mysqldump` pour sauvegarder la base de données dans un fichier SQL. Nous créons un script nommé `backup_mysql.sh` avec le contenu suivant :

```

#!/bin/bash

# Répertoire de sauvegarde
backup_dir="/"

# Nom du fichier de sauvegarde
backup_file="mysql_backup_$(date +%Y-%m-%d').sql"

# Commande de sauvegarde
mysqldump -u utilisateur -pmot_de_passe nom_de_la_base >
"$backup_dir/$backup_file"

# Compression de la sauvegarde
gzip "$backup_dir/$backup_file"

```

Dans ce script, nous utilisons le répertoire racine ('/') comme emplacement de sauvegarde. Assurez-vous d'avoir les permissions appropriées pour écrire dans ce répertoire.

Ensuite, pour PostgreSQL, nous utilisons la commande 'pg_dump' pour sauvegarder la base de données dans un fichier SQL. Nous créons un script nommé 'backup-postgresql.sh' avec le contenu suivant :

```

#!/bin/bash

# Répertoire de sauvegarde
backup_dir="/"

# Nom du fichier de sauvegarde
backup_file="postgresql_backup_$(date +%Y-%m-%d').sql"

# Commande de sauvegarde
pg_dump -U utilisateur -f "$backup_dir/$backup_file" db_tp2

```

Dans ce script, nous utilisons également le répertoire racine ('/') comme emplacement de sauvegarde.

Enfin, nous planifions l'exécution de ces scripts de sauvegarde en ajoutant des entrées à la crontab. Par exemple, pour exécuter la sauvegarde MySQL tous les jours à minuit, nous ajoutons la ligne suivante à la crontab :

```
0 0 * * * /backup_mysql.sh
```

De même, pour la sauvegarde PostgreSQL, nous ajoutons la ligne suivante à la crontab :

```
0 0 * * * /backup_postgresql.sh
```

Ainsi, nos bases de données MySQL et PostgreSQL seront sauvegardées régulièrement à la racine du système selon la planification définie.

33 Mise en place des règles de filtrage IPTables

Pour renforcer la sécurité de notre installation, nous mettons en place une matrice de filtrage iptables avec des règles spécifiques pour autoriser ou bloquer le trafic réseau selon nos besoins. Voici les étapes que nous suivons pour sécuriser notre serveur :

Tout d'abord, nous supprimons toutes les règles pré-existantes en utilisant la commande suivante dans le terminal :

```
sudo iptables -F
```

Ensuite, nous autorisons l'interface loopback pour permettre les communications internes au serveur en utilisant la commande suivante :

```
sudo iptables -A INPUT -i lo -j ACCEPT
```

Nous définissons la politique par défaut en bloquant tout le trafic entrant et sortant sauf les connexions établies et les requêtes de réponse associées :

```
sudo iptables -P INPUT DROP
sudo iptables -P FORWARD DROP
sudo iptables -P OUTPUT ACCEPT
```

Nous bloquons l'accès d'un autre groupe à notre serveur en autorisant uniquement le trafic provenant de certaines adresses IP ou plages d'adresses spécifiques :

```
sudo iptables -A INPUT -s adresse_IP -j ACCEPT
```

Nous sécurisons le serveur en ouvrant uniquement les ports strictement nécessaires pour les services que nous utilisons, tels que HTTP, HTTPS et SSH :

```
sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

Nous permettons le dialogue entre le client et le serveur en autorisant les réponses aux connexions établies depuis le serveur :

```
sudo iptables -A INPUT -m state --state RELATED,ESTABLISHED -j
ACCEPT
```

Enfin, nous sécurisons le client en ouvrant uniquement les ports strictement nécessaires pour les services qu'il utilise :

```
sudo iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT
sudo iptables -A OUTPUT -p tcp --dport 443 -j ACCEPT
sudo iptables -A OUTPUT -p tcp --dport 22 -j ACCEPT
```

Pour valider notre travail, nous nous connectons sur les sites de nos collègues pour vérifier l'accessibilité et nous utilisons éventuellement l'outil nmap pour scanner les ports ouverts sur notre routeur et nous assurer qu'ils correspondent à nos règles de filtrage iptables.

34 Automatisation du routage et de la translation d'adresses au démarrage du serveur

Pour automatiser le processus de routage et de filtrage (translation d'adresses) au démarrage de notre serveur, nous ajoutons les scripts correspondants aux scripts de démarrage système.

Tout d'abord, pour le routage, nous créons un script nommé 'route.sh' contenant les commandes de routage que nous souhaitons exécuter au démarrage. Par exemple, pour ajouter une route statique, nous pouvons utiliser les commandes suivantes :

```
#!/bin/bash

# Ajouter une route statique
ip route add network/subnet via gateway
```

Nous rendons ce script exécutable en utilisant la commande suivante :

```
chmod +x route.sh
```

Ensuite, nous plaçons ce script dans le répertoire des scripts de démarrage, par exemple '/etc/init.d/', et nous le configurons pour qu'il s'exécute au démarrage du système en utilisant la commande suivante :

```
sudo update-rc.d route.sh defaults
```

De même, pour le filtrage (translation d'adresses) avec iptables, nous créons un script nommé 'iptables.sh' contenant les règles de filtrage que nous souhaitons appliquer au démarrage. Par exemple, pour configurer la translation d'adresses (NAT), nous pouvons utiliser les commandes suivantes :

```
#!/bin/bash

# Activer le routage IPv4
echo 1 > /proc/sys/net/ipv4/ip_forward

# Translation d'adresses (NAT)
iptables -t nat -A POSTROUTING -o interface -j MASQUERADE
```

Nous rendons également ce script exécutable et le plaçons dans le répertoire des scripts de démarrage. Ensuite, nous le configurons pour qu'il s'exécute au démarrage du système de la même manière que pour le script de routage.

En ajoutant ces scripts aux scripts de démarrage, nous assurons que les commandes de routage et de filtrage (translation d'adresses) sont automatiquement exécutées à chaque démarrage du serveur, ce qui garantit la cohérence et la stabilité de notre configuration réseau.

35 Autorisation spécifique pour le webmaster : Modification de la configuration d'Apache et relance du serveur avec sudo

Pour accorder des privilèges spécifiques à certains utilisateurs, tels que le webmaster, pour modifier la configuration d'Apache et relancer le serveur, nous configurons sudo de manière appropriée.

Tout d'abord, nous éditons le fichier sudoers en utilisant la commande suivante dans le terminal :

```
sudo visudo
```

Ensuite, nous ajoutons une entrée permettant au webmaster (remplacez "webmaster" par le nom d'utilisateur approprié) d'exécuter des commandes spécifiques liées à Apache sans mot de passe. Par exemple, pour autoriser le webmaster à modifier la configuration d'Apache et relancer le serveur, nous ajoutons les lignes suivantes à la fin du fichier sudoers :

```
webmaster ALL=(ALL) NOPASSWD: /usr/sbin/apache2ctl
```

Cela permettra au webmaster d'exécuter la commande 'apache2ctl' en tant que superutilisateur sans saisir de mot de passe.

Ensuite, nous informons le webmaster des commandes qu'il peut exécuter pour modifier la configuration d'Apache et relancer le serveur. Par exemple, pour modifier le fichier de configuration d'Apache, il peut utiliser la commande suivante :

```
sudo nano /etc/apache2/apache2.conf
```

Et pour relancer le serveur Apache après avoir effectué les modifications, il peut utiliser la commande suivante :

```
sudo apache2ctl restart
```

Ainsi, en accordant ces privilèges au webmaster via sudo, nous permettons à cet utilisateur de modifier la configuration d'Apache et de relancer le serveur de manière sécurisée et contrôlée.

36 Conclusion TP2

Dans le cadre de ce TP, nous avons couvert un large éventail de tâches liées à la configuration et à la gestion d'un serveur web. De la création de comptes utilisateurs à l'installation de divers logiciels et à la sécurisation du serveur, chaque étape a été soigneusement détaillée pour assurer un environnement fonctionnel et sécurisé. Nous avons créé des comptes utilisateurs avec des répertoires personnels et des accès configurés, mis en place des VirtualHosts pour héberger plusieurs sites web avec des répertoires racine distincts, installé et configuré des bases de données MySQL et PostgreSQL avec des autorisations appropriées, et installé divers logiciels et outils comme GParted, MariaDB et PHP. De plus, nous avons mis en place des sauvegardes régulières des bases de données, renforcé la sécurité du serveur avec des règles de filtrage iptables, et automatisé le routage et la translation d'adresses au démarrage. En accordant des autorisations spécifiques à certains utilisateurs, tels que le webmaster, pour modifier la configuration d'Apache en toute sécurité, nous avons réussi à configurer un environnement de développement web diversifié et sécurisé, offrant une plateforme stable pour héberger plusieurs utilisateurs et projets.

37 Introduction TP 3

La mise en place d'un domaine Samba/LDAP constitue une étape essentielle dans la gestion des ressources et des utilisateurs au sein d'une entreprise. Ce rapport présente le processus de configuration et d'implémentation de ces services sur notre infrastructure, en utilisant notre machine mise à disposition. L'objectif principal de ce travail est de constituer un annuaire de l'entreprise, permettant l'accès et l'authentification des utilisateurs, ainsi que le partage sécurisé des ressources.

Dans cette introduction, nous allons aborder les étapes clés de cette mise en place, notamment la définition du DIT (Directory Information Tree), l'installation des paquets nécessaires, la configuration des services LDAP et Samba, la création des utilisateurs et des groupes, et enfin la validation de l'installation.

Ce rapport détaillera chaque étape du processus, en fournissant des explications claires et des instructions précises pour chaque configuration. Nous mettrons également en évidence les points importants à prendre en compte, les éventuels défis rencontrés et les solutions proposées. Enfin, nous aborderons les points bonus, offrant des possibilités d'amélioration et d'extension de notre infrastructure.

38 Définition du DIT (Directory Information Tree)

La structuration de l'annuaire LDAP, représentée par le DIT (Directory Information Tree), constitue une étape cruciale dans la mise en place du domaine Samba/LDAP pour notre agence à Lyon. Cette

section détaille la manière dont nous avons défini la hiérarchie de notre annuaire, en utilisant des composants adaptés à notre environnement spécifique.

La structure du DIT suivante a été élaborée pour répondre aux besoins de notre agence :

- 'DC=agence, DC=lyon' - 'OU=Users': Cette unité organisationnelle rassemble les informations relatives aux utilisateurs de notre agence à Lyon, permettant une gestion centralisée des comptes utilisateurs. - 'OU=Groups': Ici sont répertoriés les différents groupes d'utilisateurs, facilitant la gestion des autorisations et des accès aux ressources partagées. - 'OU=Computers': Cette unité organisationnelle contient les données concernant les ordinateurs et équipements informatiques de notre agence, permettant une gestion efficace du parc informatique.

En utilisant cette structure bien définie, nous sommes en mesure de créer un annuaire LDAP organisé et fonctionnel, adapté à notre environnement spécifique à Lyon.

39 Installation des paquets et des schémas LDAP

Pour mettre en place notre domaine Samba/LDAP, il est nécessaire d'installer les paquets logiciels requis ainsi que les schémas LDAP nécessaires à la configuration de nos attributs. Cette section détaille les étapes suivies pour réaliser cette installation :

1. Installation des paquets logiciels : Nous avons utilisé les gestionnaires de paquets de notre système pour installer les logiciels nécessaires à la mise en place du domaine Samba/LDAP. Parmi les paquets essentiels, nous avons notamment installé 'samba', 'ldap-utils', 'smbldap-tools' et 'libnss-ldap'.

2. Téléchargement des schémas LDAP : Nous avons récupéré les schémas LDAP requis pour notre configuration. Ces schémas définissent la structure et les attributs de notre annuaire LDAP, et sont nécessaires pour garantir la compatibilité et la fonctionnalité de notre système.

3. Intégration des schémas : Une fois les schémas téléchargés, nous les avons intégrés à notre serveur LDAP en suivant les instructions spécifiques à notre environnement et à nos besoins. Cela peut impliquer la copie des fichiers de schémas dans un répertoire spécifique, suivi d'une étape de configuration pour les activer dans notre système LDAP.

En suivant ces étapes, nous avons pu installer avec succès les paquets logiciels requis et intégrer les schémas LDAP nécessaires à la configuration de notre domaine Samba/LDAP. Cette préparation est essentielle pour garantir le bon fonctionnement et la compatibilité de notre système.

40 Paramétrage du service LDAP

La configuration du service LDAP, définie dans le fichier '/etc/ldap/slapd.conf', est une étape cruciale dans la mise en place de notre domaine Samba/LDAP. Cette section décrit les actions entreprises pour configurer ce service et garantir son bon fonctionnement :

1. Modification du fichier slapd.conf : Nous avons édité le fichier '/etc/ldap/slapd.conf' pour définir les paramètres de configuration nécessaires à notre environnement LDAP. Ces paramètres incluent notamment la définition du schéma LDAP, les accès à la base de données, les contraintes de sécurité, etc. Nous avons veillé à configurer ces paramètres conformément aux besoins spécifiques de notre système.

2. Arrêt du service slapd : Avant de procéder à toute modification du fichier de configuration ou à la suppression du répertoire '/etc/ldap/slapd.d', nous avons arrêté le service slapd pour éviter toute interruption ou corruption des données en cours d'opération.

3. Suppression du répertoire slapd.d : Une fois le service slapd arrêté, nous avons procédé à la suppression du répertoire '/etc/ldap/slapd.d', qui contient les fichiers de configuration générés automatiquement par OpenLDAP. Cette étape est nécessaire pour permettre la régénération des fichiers de configuration en fonction des modifications apportées au fichier 'slapd.conf'.

4. Redémarrage du service slapd : Une fois le répertoire '/etc/ldap/slapd.d' supprimé, nous avons redémarré le service slapd pour appliquer les modifications de configuration. Cela a permis de reconstruire les fichiers de configuration à partir du fichier 'slapd.conf' modifié, rendant ainsi le service LDAP opérationnel avec les nouveaux paramètres configurés.

En suivant ces étapes, nous avons pu paramétrer avec succès le service LDAP selon nos besoins spécifiques, en veillant à respecter les bonnes pratiques de configuration et de maintenance du système.

41 Paramétrage de l'exploitation des données de l'annuaire LDAP

Afin d'exploiter les données de notre annuaire LDAP sur notre machine, nous avons configuré le paquet 'libnss-ldap'. Cette section détaille les étapes suivies pour réaliser cette configuration :

1. Installation du paquet libnss-ldap : Nous avons utilisé le gestionnaire de paquets de notre système pour installer le paquet 'libnss-ldap'. Ce paquet permet à notre système d'utiliser le service LDAP pour la résolution des noms d'utilisateurs, de groupes, etc.
2. Configuration du fichier '/etc/nsswitch.conf' : Nous avons édité le fichier '/etc/nsswitch.conf' pour spécifier l'utilisation de LDAP pour la résolution des noms d'utilisateurs, de groupes, etc. Dans ce fichier, nous avons configuré les entrées telles que 'passwd', 'group', 'shadow', etc., pour qu'elles utilisent LDAP en plus des sources de données locales.
3. Configuration du fichier '/etc/ldap.conf' : Nous avons également configuré le fichier '/etc/ldap.conf' pour définir les paramètres de connexion au serveur LDAP, tels que l'adresse du serveur, le port, les informations d'authentification, etc. Ces paramètres permettent à notre machine d'établir une connexion avec le serveur LDAP et de récupérer les données nécessaires.

En suivant ces étapes, nous avons configuré notre machine pour exploiter les données de notre annuaire LDAP à l'aide du paquet 'libnss-ldap'. Cette configuration permet à notre système d'accéder aux informations d'authentification et aux autres données stockées dans l'annuaire LDAP, facilitant ainsi l'authentification des utilisateurs et l'accès aux ressources partagées.

42 Installation et configuration des smbldap-tools

L'installation et la configuration des smbldap-tools sont des étapes essentielles pour la gestion des utilisateurs et des groupes dans notre domaine Samba/LDAP. Cette section détaille les actions entreprises pour installer et configurer ces outils en fonction de notre DIT (Directory Information Tree) :

1. Installation des smbldap-tools : Nous avons utilisé le gestionnaire de paquets de notre système pour installer les smbldap-tools version 0.9.9-1 ou supérieur. Ces outils sont nécessaires pour la gestion des utilisateurs, des groupes et des autres objets dans notre domaine Samba/LDAP.
2. Configuration des fichiers smbldap.conf et smbldap_bind.conf : Nous avons édité les fichiers de configuration '/etc/smbldap-tools/smbldap.conf' et '/etc/smbldap-tools/smbldap_bind.conf' pour les adapter à notre environnement LDAP. Dans ces fichiers, nous avons spécifié les paramètres de connexion au serveur LDAP, tels que l'adresse du serveur, le port, les informations d'authentification, etc. De plus, nous avons configuré d'autres options spécifiques à notre infrastructure, telles que le SID (Security Identifier) de notre domaine Samba/LDAP.
3. Renseignement du SID de notre domaine : Nous avons veillé à renseigner correctement le SID de notre domaine dans le fichier de configuration smbldap.conf. Le SID est un identifiant unique qui permet à Samba de différencier notre domaine des autres domaines présents sur le réseau. En le configurant correctement, nous avons assuré la cohérence et la sécurité de notre environnement Samba/LDAP.

En suivant ces étapes, nous avons installé et configuré les smbldap-tools en fonction de notre DIT, ce qui nous permet de gérer efficacement les utilisateurs, les groupes et les autres objets dans notre domaine Samba/LDAP. Cette configuration garantit une gestion centralisée et sécurisée de notre infrastructure, facilitant ainsi la gestion des ressources et des autorisations.

43 Création des ressources à partager et configuration du service Samba

La création des ressources à partager et la configuration du service Samba sont des étapes cruciales pour permettre l'accès aux ressources partagées sur notre réseau. Voici comment nous avons procédé pour accomplir cette tâche :

1. Création des répertoires à partager : Nous avons créé les répertoires sur notre système de fichiers que nous souhaitons partager avec les utilisateurs du réseau. Ces répertoires peuvent contenir des fichiers, des dossiers ou d'autres ressources que nous voulons rendre accessibles aux utilisateurs du domaine.

2. Édition du fichier de configuration `smb.conf` : Nous avons édité le fichier de configuration principal de Samba, `/etc/samba/smb.conf`, pour définir les paramètres de partage pour nos ressources. Dans ce fichier, nous avons spécifié les différents paramètres pour chaque partage, tels que le nom du partage, le chemin d'accès au répertoire partagé, les autorisations d'accès, etc.

3. Configuration des autorisations et des paramètres de sécurité : Nous avons également configuré les autorisations et les paramètres de sécurité pour chaque partage dans le fichier `smb.conf`. Cela inclut la spécification des autorisations d'accès aux utilisateurs et aux groupes, la définition des modes de sécurité (par exemple, le mode de sécurité utilisateur, le mode de sécurité partage), la gestion des autorisations de lecture, d'écriture et d'exécution, etc.

4. Test et validation du partage : Une fois que nous avons configuré les paramètres de partage dans le fichier `smb.conf`, nous avons testé et validé chaque partage en redémarrant le service Samba et en vérifiant l'accès aux ressources partagées à partir d'autres machines du réseau. Cela nous a permis de nous assurer que les utilisateurs pouvaient accéder aux ressources partagées conformément aux autorisations et aux paramètres configurés.

En suivant ces étapes, nous avons créé et configuré avec succès les ressources à partager sur notre réseau à l'aide du service Samba. Cette configuration nous permet de fournir un accès sécurisé et centralisé aux ressources partagées, ce qui facilite la collaboration et le partage de fichiers entre les utilisateurs du domaine.

44 Peuplement de l'annuaire avec `smbldap-populate`

Pour peupler notre annuaire avec `smbldap-populate`, une série d'étapes sont nécessaires pour assurer que notre annuaire LDAP soit correctement configuré et que les utilisateurs et groupes soient ajoutés avec succès. Voici comment nous avons procédé :

1. Configuration préalable de `smbldap-tools` : Avant de pouvoir utiliser `smbldap-populate`, nous avons d'abord configuré les fichiers de paramètres `smbldap.conf` et `smbldap_bind.conf` dans `/etc/smbldap-tools/` en fonction de notre environnement LDAP. Ces fichiers contiennent des informations telles que les noms de domaine, les mots de passe d'administration LDAP, etc.

2. Exécution de `smbldap-populate` : Une fois que `smbldap-tools` a été configuré, nous avons exécuté la commande `smbldap-populate` pour peupler notre annuaire LDAP avec des données d'exemple. Cette commande crée généralement une structure de base pour les utilisateurs, les groupes et les autres objets dans l'annuaire LDAP.

3. Validation du peuplement : Après l'exécution de `smbldap-populate`, nous avons vérifié l'annuaire LDAP pour nous assurer que les utilisateurs, les groupes et autres objets ont été ajoutés avec succès. Nous avons examiné les entrées LDAP à l'aide d'outils tels que `ldapsearch` pour confirmer que les données ont été peuplées correctement.

4. Test de l'authentification et de l'accès aux ressources : Après le peuplement de l'annuaire, nous avons testé l'authentification des utilisateurs et leur accès aux ressources partagées sur notre réseau. Cela nous a permis de vérifier que les utilisateurs pouvaient se connecter avec leurs identifiants LDAP et accéder aux ressources autorisées en fonction de leurs permissions.

En suivant ces étapes, nous avons réussi à peupler notre annuaire LDAP avec des données d'exemple à l'aide de `smbldap-populate`. Cette étape est essentielle pour assurer que notre annuaire LDAP est opérationnel et prêt à être utilisé pour l'authentification des utilisateurs et la gestion des ressources sur notre réseau.

45 Conclusion TP3

La mise en place d'un domaine Samba/LDAP constitue une étape cruciale dans le développement d'une infrastructure informatique fiable et sécurisée pour toute entreprise. Ce processus, que nous avons exploré en détail tout au long de ce TP, nous a permis de saisir pleinement l'importance de l'authentification centralisée et du partage de ressources au sein d'un environnement réseau.

Au fil des différentes étapes que nous avons suivies, nous avons réussi à déployer avec succès les services LDAP et Samba. En définissant une structure d'annuaire adaptée à notre entreprise et en configurant les paramètres appropriés, nous avons établi un environnement où les utilisateurs peuvent être authentifiés de manière sécurisée et où les ressources peuvent être partagées de manière efficace.

Cette implémentation offre une multitude d'avantages, notamment une gestion centralisée des identités et des accès, une amélioration de la sécurité grâce à l'application de politiques d'accès et de contrôle, ainsi qu'une facilité de gestion des ressources partagées. De plus, la solution Samba/LDAP offre une grande flexibilité et extensibilité, permettant à notre infrastructure informatique de s'adapter aux besoins évolutifs de notre entreprise.

En conclusion, ce TP nous a permis de maîtriser les concepts essentiels associés à la configuration d'un domaine Samba/LDAP et de développer les compétences nécessaires pour concevoir, configurer et administrer un tel environnement. Cette expérience enrichissante sera certainement précieuse dans notre parcours professionnel, en nous dotant des connaissances et des compétences requises pour relever les défis complexes de l'administration système et de la gestion des réseaux.