

Rapport de projet: DHCP / RELAY / DNS / WEB

DIAWARA NANA
Module : Administration Linux

January 20, 2025

Contents

1	Introduction	2
2	DHCP, DNS, WEB, RELAY	2
2.1	DHCP (Dynamic Host Configuration Protocol)	2
2.2	DNS (Domain Name System)	2
2.3	Web (World Wide Web)	2
2.4	Relay (Relais)	2
3	Objectif du TP	2
4	Configuration	3
4.1	Serveur DHCP	3
4.2	Relay DHCP	4
4.3	Serveur DNS	5
4.4	Serveur Web	7
5	Conclusion	8

1 Introduction

Dans un réseau informatique, plusieurs technologies collaborent pour assurer une communication fluide et une gestion optimale des ressources. Parmi elles, DHCP, DNS, le Web, et les relais jouent des rôles essentiels. Comprendre leur fonctionnement est crucial pour administrer un réseau efficacement.

2 DHCP, DNS, WEB, RELAY

2.1 DHCP (Dynamic Host Configuration Protocol)

Le DHCP est un protocole utilisé pour automatiser l'attribution des adresses IP, des passerelles par défaut et des serveurs DNS aux périphériques d'un réseau. Il simplifie la gestion des adresses IP en évitant la configuration manuelle et en réduisant les conflits d'adresses.

2.2 DNS (Domain Name System)

Le DNS est le système qui traduit les noms de domaine lisibles par les humains (comme `www.google.com`) en adresses IP compréhensibles par les machines. Il agit comme un annuaire téléphonique pour Internet, facilitant l'accès aux services en ligne et rendant les réseaux plus conviviaux.

2.3 Web (World Wide Web)

Le Web représente l'ensemble des ressources accessibles sur Internet via des navigateurs. Ces ressources, comme les sites Web et les applications Web, sont rendues disponibles grâce à des technologies comme HTTP/HTTPS et les serveurs Web.

2.4 Relay (Relais)

En réseau, un relay est un dispositif ou une configuration qui transmet des paquets ou des données entre différents segments de réseau. Un exemple courant est le relais DHCP, qui permet à des clients situés sur un réseau différent de recevoir des configurations d'un serveur DHCP.

3 Objectif du TP

L'objectif de ce TP est de maîtriser la configuration et le déploiement des serveurs essentiels au fonctionnement d'un réseau sur une machine Ubuntu. Plus précisément, il s'agit d'apprendre à configurer :

- **Serveur DHCP** : Configurer un serveur DHCP pour automatiser l'attribution des adresses IP, des passerelles par défaut et des serveurs DNS aux clients du réseau, tout en optimisant la gestion des ressources IP.

- **DHCP Relay** : Implémenter un relais DHCP pour permettre aux clients situés sur des sous-réseaux différents d'obtenir leurs configurations depuis un serveur DHCP central.
- **Serveur DNS** : Mettre en place un serveur DNS capable de traduire les noms de domaine en adresses IP, et inversement.
- **Serveur Web** : Installer et configurer un serveur Web pour héberger des pages ou des applications accessibles via des navigateurs.

4 Configuration

4.1 Serveur DHCP

1. Mise à jour `sudo apt update` :
2. Installation du serveur DHCP sur une machine Ubuntu :

```
ana@nana-VMware-Virtual-Platform:~$ sudo apt install isc-dhcp-relay
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des listes de paquets... Fait
```

3. Première chose à paramétrer est l'interface d'écoute du serveur DHCP :

```
GNU nano 7.2 /etc/default/isc-dhcp-server *
Defaults for isc-dhcp-server (sourced by /etc/init.d/isc-dh

Path to dhcpd's config file (default: /etc/dhcp/dhcpd.conf)
DHCPDv4_CONF=/etc/dhcp/dhcpd.conf
DHCPDv6_CONF=/etc/dhcp/dhcpd6.conf

Path to dhcpd's PID file (default: /var/run/dhcpd.pid).
DHCPDv4_PID=/var/run/dhcpd.pid
DHCPDv6_PID=/var/run/dhcpd6.pid

Additional options to start dhcpd with.
Don't use options -cf or -pf here; use DHCPD_CONF/ DH
OPTIONS=""

On what interfaces should the DHCP server (dhcpd) serve DHCP
Separate multiple interfaces with spaces, e.g. "eth0
INTERFACESv4="ens33"
INTERFACESv6=""
```

4. Configuration du sous-réseau dans le fichier `/etc/dhcp/dhcpd.conf` :

```

GNU nano 7.2 /etc/dhcp/dhcpd.conf *
#shared-network 224-29 {
#   subnet 10.17.224.0 netmask 255.255.255.0 {
#       option routers rtr-224.example.org;
#   }
#   subnet 10.0.29.0 netmask 255.255.255.0 {
#       option routers rtr-29.example.org;
#   }
#   pool {
#       allow members of "foo";
#       range 10.17.224.10 10.17.224.250;
#   }
#   pool {
#       deny members of "foo";
#       range 10.0.29.10 10.0.29.230;
#   }
#}
subnet 192.168.1.0 netmask 255.255.255.0 {
    range 192.168.1.2 192.168.1.253;
}

```

5. Configuration manuelle de l'adresse IP du serveur :



6. Redémarrer le service DHCP :

7. Vérifier qu'il est actif `sudo systemctl status isc-dhcp-server` :

4.2 Relay DHCP

1. Mise à jour `sudo apt update` :

2. Installation du serveur de relai :

```

ana@nana-VMware-Virtual-Platform:~$ sudo apt install isc-dhcp-relay
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait

```

3. Modifier le fichier de configuration principal `/etc/default/isc-dhcp-relay` :

```

GNU nano 7.2 /etc/default/isc-dhcp-relay
SERVERS="192.168.1.2"
INTERFACES=""
OPTIONS=""

```

4. Redémarrer le service pour appliquer les changements :

4.3 Serveur DNS

1. Mise à jour des paquets sur la machine `sudo apt update` :
2. Installer bind et les outils associés :

```
nana@nana-VMware-Virtual-Platform:~$ sudo apt install bind9 bind9utils bind9-doc dnstools
[sudo] Mot de passe de nana :
```

3. Configurer le fichier `named.conf` :

```
GNU nano 7.2 /etc/bind/named.conf.local
//
// Do any local configuration here
//
// Consider adding the 1918 zones here, if they are not used in yo
// organization
//include "/etc/bind/zones.rfc1918";
//DEFINITION DE LA ZONE DIRECTE
zone "mydomain.local"{
type master;
file "/etc/bind/db.mydomain.local";//FICHIER DE ZONE POUR LE DOMA
};

//DEFINITION DE LA ZONE INVERSE
zone "1.168.192.in-addr.arpa" {
type master;
file "/etc/bind/db.192";//FICHIER DE ZONE INVERSE POUR LES IP 192.
};
```

4. Créer le fichier de zone:

```
GNU nano 7.2 /etc/bind/db.mydomain.local
$TTL 86400
@ IN SOA ns1.mydomain.local. admin.mydomain.local. (
2025010601 ; Serial (année, mois, jour, numéro de révision)
3600 ; Refresh (1 heure)
1800 ; Retry (30 minutes)
1209600 ; Expire (2 semaines)
86400 ) ; Minimum TTL (1 jour)

; Définition des serveurs de noms
@ IN NS ns1.mydomain.local.
@ IN NS ns2.mydomain.local.

; Adresses des serveurs de noms
ns1 IN A 192.168.1.2
ns2 IN A 192.168.1.3

; Enregistrement des hôtes
www IN A 192.168.1.2
mail IN A 192.168.1.2
ftp IN A 192.168.1.30

; Enregistrements MX pour le mail
@ IN MX 10 mail.mydomain.local.
```

5. Créer le fichier de zone inverse:

```

GNU nano 7.2 /etc/bind/db.192
$TTL      86400
@          IN      SOA      ns1.mydomain.local. admin.mydomain.local. (
                                2025010601 ; Serial
                                3600        ; Refresh
                                1800        ; Retry
                                1209600     ; Expire
                                86400 )     ; Minimum TTL

; Définition des serveurs de noms
@          IN      NS       ns1.mydomain.local.
@          IN      NS       ns2.mydomain.local.

; Enregistrements PTR (pour la résolution inverse)
2          IN      PTR      ns1.mydomain.local.
3          IN      PTR      ns2.mydomain.local.
2          IN      PTR      www.mydomain.local.
2          IN      PTR      mail.mydomain.local.
30         IN      PTR      ftp.mydomain.local.

```

6. Ajouter le DNS au fichier `/etc/dhcp/dhcpd.conf` :

```

GNU nano 7.2 /etc/dhcp/dhcpd.conf *
# You can declare a class of clients and then do address allocation
# based on that.  The example below shows a case where all clients
# in a certain class get addresses on the 10.17.224/24 subnet,
# other clients get addresses on the 10.0.29/24 subnet.

#class "foo" {
#  match if substring (option vendor-class-identifier, 0, 4) = "foo";
#}

#shared-network 224-29 {
#  subnet 10.17.224.0 netmask 255.255.255.0 {
#    option routers rtr-224.example.org;
#  }
#  subnet 10.0.29.0 netmask 255.255.255.0 {
#    option routers rtr-29.example.org;
#  }
#  pool {
#    allow members of "foo";
#    range 10.17.224.10 10.17.224.250;
#  }
#  pool {
#    deny members of "foo";
#    range 10.0.29.10 10.0.29.230;
#  }
#}
subnet 192.168.1.0 netmask 255.255.255.0 {
  range 192.168.1.3 192.168.1.253;
  option routers 192.168.1.2; # Passerelle
  option domain-name-servers 192.168.1.2; # DNS
}

```

7. Vérifier et Tester la configuration :

```

nana@nana-VMware-Virtual-Platform:~$ sudo named-checkconf
nana@nana-VMware-Virtual-Platform:~$ sudo named-checkzone mydomain.
local /etc/bind/db.mydomain.local
sudo named-checkzone 1.168.192.in-addr.arpa /etc/bind/db.192
zone mydomain.local/IN: loaded serial 2025010601
OK
zone 1.168.192.in-addr.arpa/IN: loaded serial 2025010601
OK
nana@nana-VMware-Virtual-Platform:~$ sudo systemctl restart bind9
nana@nana-VMware-Virtual-Platform:~$ sudo systemctl enable named
Synchronizing state of named.service with SysV service script with
/usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable named
nana@nana-VMware-Virtual-Platform:~$

```

```
nana@nana-VMware-Virtual-Platform:~$ sudo nano /etc/resolv.conf
[sudo] Mot de passe de nana :
nana@nana-VMware-Virtual-Platform:~$ nslookup www.mydomain.local
Server:      192.168.1.2
Address:     192.168.1.2#53

Name:   www.mydomain.local
Address: 192.168.1.2

nana@nana-VMware-Virtual-Platform:~$ nslookup -x 192.168.1.2
*** Invalid option: x
2.1.168.192.in-addr.arpa    name = mail.mydomain.local.
2.1.168.192.in-addr.arpa    name = www.mydomain.local.
2.1.168.192.in-addr.arpa    name = ns1.mydomain.local.
```

4.4 Serveur Web

1. Installer apache:

```
nana@nana-VMware-Virtual-Platform:~$ sudo apt install apache2
Lecture des listes de paquets... Fait
Construction de l'arbre de dépendances... Fait
Sélection de packages à installer... Fait
Résolution des dépendances... Fait
Les fichiers suivants vont être installés :
  apache2
0 packages à télécharger, 0 octets à télécharger, 0 octets de fichiers supplémentaires à télécharger, 0 octets de fichiers supplémentaires à télécharger.
Préparation de l'installation de apache2... Fait
Installation de apache2... Fait
0 octets de fichiers supplémentaires à télécharger, 0 octets de fichiers supplémentaires à télécharger.
Après la mise à jour, 1 fichier supplémentaire sera installé.
0 octets de fichiers supplémentaires à télécharger, 0 octets de fichiers supplémentaires à télécharger.
Préparation de l'installation de apache2... Fait
Installation de apache2... Fait
0 octets de fichiers supplémentaires à télécharger, 0 octets de fichiers supplémentaires à télécharger.
Après la mise à jour, 1 fichier supplémentaire sera installé.
0 octets de fichiers supplémentaires à télécharger, 0 octets de fichiers supplémentaires à télécharger.
```

2. Créer un fichier de configuration pour le site:

```
GNU nano 7.2 /etc/apache2/sites-available/mydomain.conf
<VirtualHost *:80>
    ServerName www.mydomain.local
    DocumentRoot /var/www/html
    <Directory /var/www/html>
        AllowOverride All
        Require all granted
    </Directory>
    ErrorLog ${APACHE_LOG_DIR}/mydomain_error.log
    CustomLog ${APACHE_LOG_DIR}/mydomain_access.log combined
</VirtualHost>
```

3. Activer le site et recharger Apache :

```
nana@nana-VMware-Virtual-Platform:~$ sudo a2ensite mydomain.conf
Enabling site mydomain.
To activate the new configuration, you need to run:
  systemctl reload apache2
nana@nana-VMware-Virtual-Platform:~$ sudo systemctl reload apache2
nana@nana-VMware-Virtual-Platform:~$ sudo chown -R www-data:www-data /var/www/html
nana@nana-VMware-Virtual-Platform:~$ sudo chmod -R 755 /var/www/html
nana@nana-VMware-Virtual-Platform:~$ sudo ufw allow 80
nana@nana-VMware-Virtual-Platform:~$ sudo ufw allow 443
nana@nana-VMware-Virtual-Platform:~$ sudo ufw reload
Les règles ont été mises à jour
Les règles ont été mises à jour (IPv6)
Les règles ont été mises à jour
Les règles ont été mises à jour (IPv6)
Pare-feu inactif (rechargement ignoré)
nana@nana-VMware-Virtual-Platform:~$
```

4. Test dans le navigateur de la machine :



5 Conclusion

À l'issue de ce TP, nous avons acquis les compétences nécessaires pour configurer et gérer les services fondamentaux d'un réseau sur une machine Ubuntu. Nous avons appris à :

- Configurer un serveur DHCP pour automatiser l'attribution des adresses IP.
- Mettre en place un relais DHCP pour permettre la communication entre des sous-réseaux et un serveur DHCP central.
- Déployer un serveur DNS pour la résolution des noms de domaine.
- Installer et configurer un serveur Web pour héberger des sites ou des applications.