

## Filière WM 2024

**Module : Administration linux avancée**



**Rapport Final**

Réalisé par

**OULOUCHOUTAM Hanane**

**Date : le 25/05/2024**

**Encadré par :**

**Amamou Ahmed**

Année Universitaire : 2023/2024

**Plan**

Introduction

DHCP Server

DNS Server

Phpmyadmin

Système de messagerie

**Introduction**

Dans le cadre de notre cours de Linux, nous avons récemment entrepris un travail pratique visant à configurer un serveur DHCP (Dynamic Host Configuration Protocol) et un serveur DNS (Domain Name System). L'objectif de cet exercice était de comprendre et de mettre en œuvre ces services fondamentaux dans un environnement réseau simulé.

Le serveur DHCP est crucial pour la gestion dynamique des adresses IP au sein du réseau, offrant une solution efficace pour l'attribution automatique des adresses IP aux périphériques clients. En parallèle, le serveur DNS assure la résolution des noms de domaine en adresses IP, facilitant ainsi la navigation sur Internet et l'accès aux ressources réseau internes.

Dans ce rapport, nous détaillerons les étapes que nous avons suivies pour configurer ces services, en mettant l'accent sur les paramètres choisis, les zones DNS configurées et les tests effectués pour valider le bon fonctionnement de nos configurations. Nous examinerons également les éventuelles difficultés rencontrées et les solutions apportées tout au long du processus.

L'expérience acquise lors de cette configuration nous a permis de mieux comprendre le rôle crucial du DHCP et du DNS dans la gestion efficace d'un réseau informatique.

# DHCP Server

**Le DHCP est un protocole essentiel pour la gestion efficace des réseaux IP.** Il simplifie la configuration des appareils, réduit les erreurs humaines et garantit une meilleure cohérence et fiabilité du réseau.

# Objectifs de Configuration

La présente configuration vise à établir un serveur DHCP capable d'attribuer dynamiquement des adresses IP aux appareils qui se connectent au réseau**.** Cette initiative s'articule autour de trois objectifs précis :

**1. Délimitation d'une plage d'adresses IP:**

Définir une plage d'adresses IP disponibles que le serveur DHCP peut attribuer aux clients.

Cette plage doit être suffisamment large pour répondre aux besoins du réseau tout en évitant les chevauchements avec d'autres réseaux.

**2. Spécification des paramètres réseau:**

Définir les paramètres réseau que le serveur DHCP transmettra aux clients, tels que :

Le masque de sous-réseau

La passerelle par défaut

Les serveurs DNS

Garantir la cohérence et la précision de ces paramètres pour une connectivité optimale des clients.

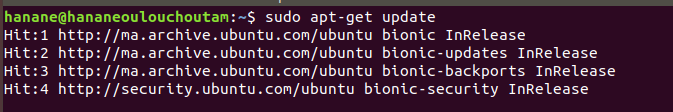
**3. Assurance d'une gestion efficace des adresses IP:**

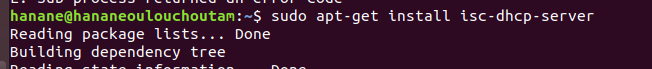
Mettre en place des mécanismes de suivi et de contrôle pour garantir une utilisation optimale des adresses IP attribuées.

Gérer les baux DHCP et les renouvellements d'adresses pour maintenir une organisation et une efficacité optimales.

# Configuration du Serveur DHCP

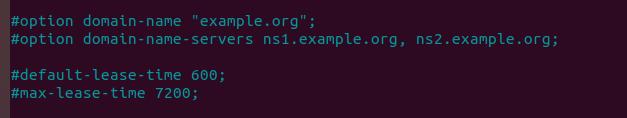
### 1. Installation du Serveur DHCP



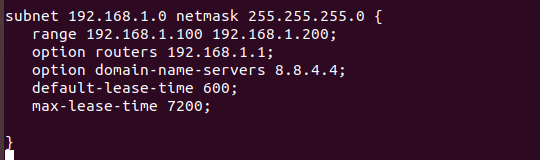


### 2. Configuration du Fichier dhcpd.conf

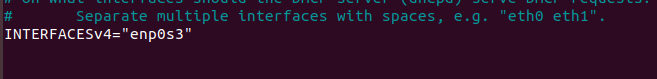




### 3. Configurer le sous-réseau et les paramètres de location



### 4. Configurer l'interface réseau

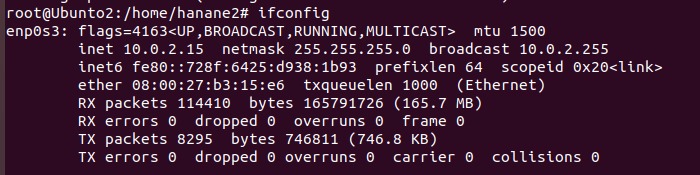


### 5. Redémarrer le serveur DHCP :

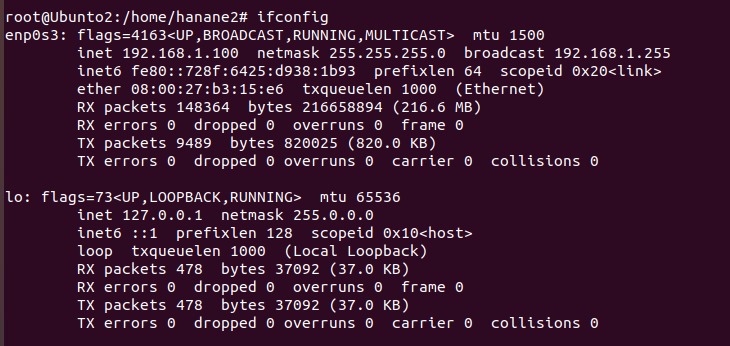
### 

### Tests de Validation

L'adresse IP du client avant le redémarrage du serveur DHCP



L'adresse IP du client après le redémarrage du serveur DHCP : 192.168.1.100



**DNS Server**

**Configuration du Serveur DNS**

### 

### 1. Installation du Serveur DNS

### 

### Nous avons installé le serveur DNS BIND9 sur Ubuntu en utilisant la commande sudo apt-get install bind9. BIND9 est maintenant prêt à être configuré pour notre domaine eidia.uemf.

### 2. Configuration du fichier de zones :

### Nous avons ouvert le fichier de configuration principal de BIND situé à /etc/bind/named.conf.local et ajouté une zone de recherche directe pour le domaine eidia.uemf. Cela permettra à BIND de gérer la résolution des noms pour notre domaine.

### 

### \* Configuration du fichier de zone pour eidia.uemf

### 

### \* Configuration de la résolution inverse

### Nous avons ajouté une zone de recherche inverse dans le fichier de configuration principal de BIND pour

### gérer la résolution inverse des adresses IP dans notre réseau. Nous avons créé le fichier de zone inverse

### /etc/bind/db.192.168.1 et ajouté les enregistrements PTR pour associer les adresses IP à leurs noms de

### domaine correspondants.

### \* Configuration du fichier de zone pour 192.168.1

### 

### 

### 

### 3. Configuration du fichier resolv :

### Nous avons modifié le fichier de configuration /etc/resolv.conf en ajoutant les informations de recherche et de serveur DNS pour notre domaine eidia.uemf. Cela permettra aux clients de résoudre les noms de domaine dans notre domaine à l'aide du serveur DNS BIND que nous avons configuré.

### 

### 4. Vérification de la configuration :

### Nous avons utilisé les commandes named-checkconf et named-checkzone pour vérifier la syntaxe de nos

### fichiers de configuration et de zone. Cela garantit que nos configurations sont correctes et prêtes à être

### utilisées par le serveur DNS BIND.

### 

### 

### 

### 5. Redémarrage du service BIND :

### Nous avons redémarré le service BIND en utilisant la commande sudo systemctl restart bind9 pour

### appliquer les nouvelles configurations que nous avons ajoutées. Cela assure que nos modifications prennent

### effet et que le serveur DNS est opérationnel avec les nouvelles configurations.

### 

### 

### 6. Test de résolution DNS :

### Nous avons utilisé la commande nslookup pour tester la résolution DNS en interrogeant notre serveur DNS

### pour résoudre le nom de domaine www.eidia.uemf. Cela nous assure que notre serveur DNS fonctionne

### correctement et peut résoudre les noms de domaine dans notre domaine eidia.uemf.

### 

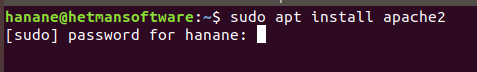
### 

### 

**Phpmyadmin**

**Configuration du Serveur DNS**

#### 1.Installation d'Apache2 :



#### 2.Vérification de l'installation :

Tester dans le navigateur : localhost

#### 3.Installation de PHP et des modules nécessaires :

#### 

#### 

#### 

#### 4.Redémarrage du service Apache :

#### 

#### 5.Installation de MySQL :

#### 

#### 

#### 6.Installation de phpMyAdmin :

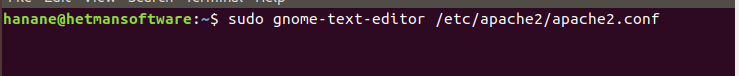
#### 

#### 7.Configuration de phpMyAdmin :

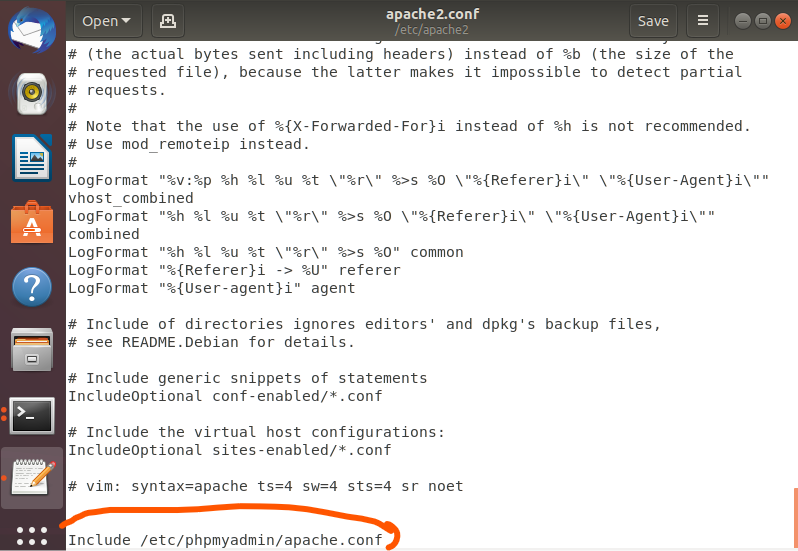
Redémarrage d'Apache



Modification du fichier de configuration d'Apache

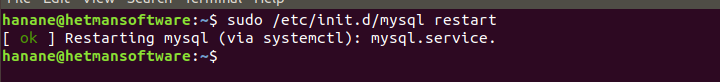


Ajouter la ligne suivante à la fin du fichier :



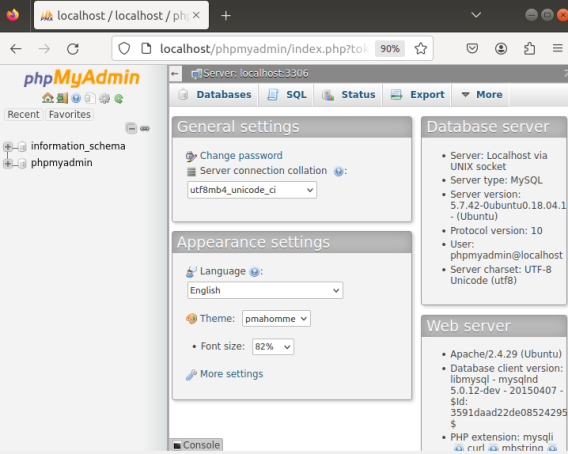
#### 8.Redémarrage des services :

-> sudo systemctl restart apache2



### Conclusion

phpMyAdmin a été installé et configuré avec succès sur le système Ubuntu. Cela permet une gestion efficace des bases de données MySQL via une interface web intuitive



**Système de messagerie**

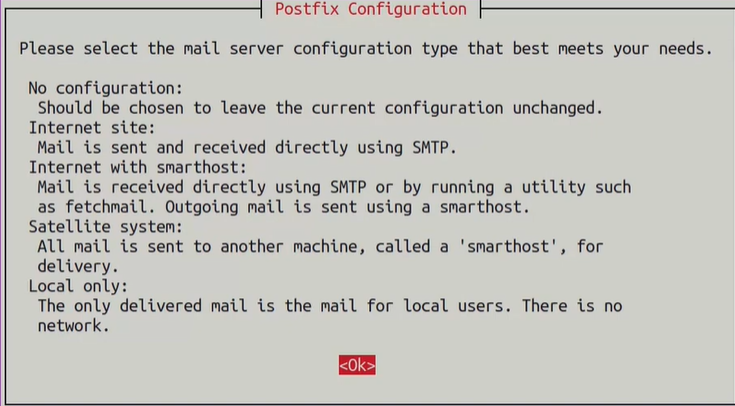
### Introduction

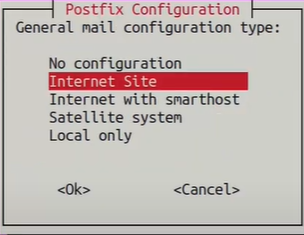
Ce chapitre détaille les étapes nécessaires pour installer et configurer un système de messagerie sur Ubuntu

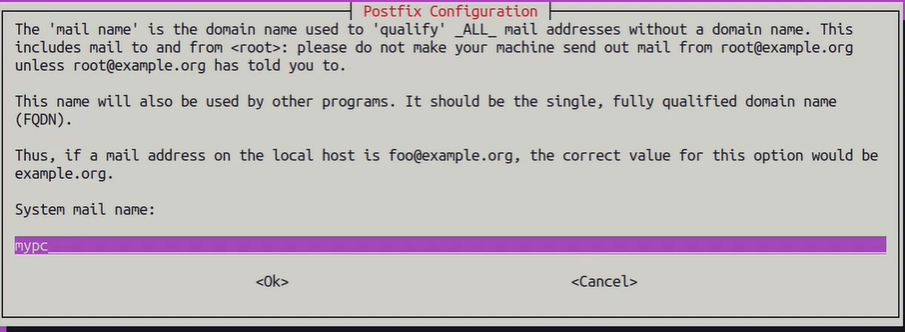
#### 1.Installation de Postfix :

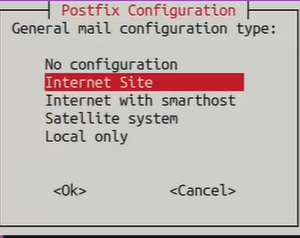
“sudo apt-get install postfix”

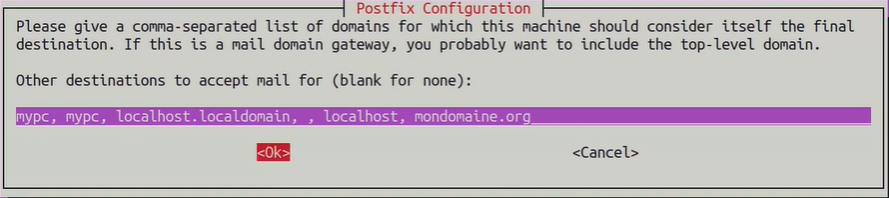
Cette commande installe Postfix, un serveur de messagerie SMTP, pour gérer l'envoi et la réception de courriels.

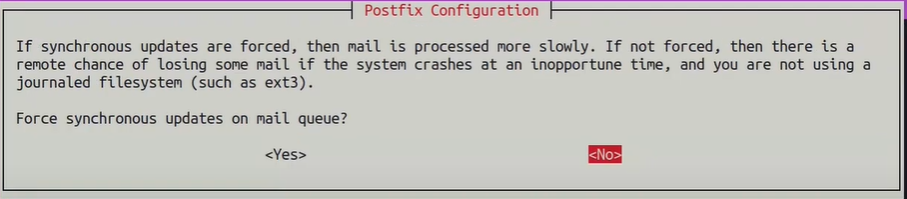


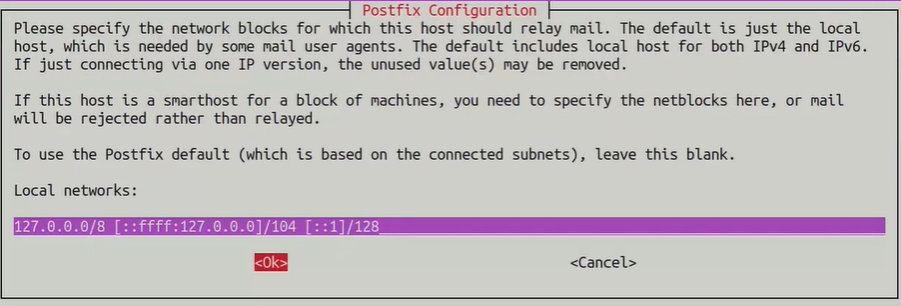


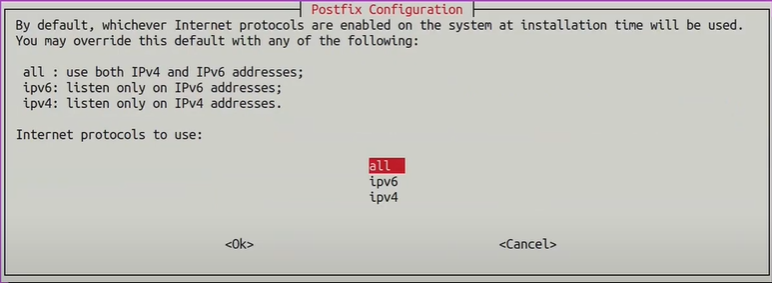












#### 2.Reconfiguration de Postfix:

#### “sudo systemctl restart postfix

#### Cette commande permet de personnaliser la configuration de Postfix après son installation, en adaptant ses paramètres au besoin.

#### 3.Redémarrage de Postfix:

#### 

#### 

Redémarre le service Postfix pour appliquer les nouvelles configurations.

#### 4.Vérification des connexions réseau:

#### 

#### 

Affiche les connexions réseau actives et les ports en écoute pour s'assurer que Postfix fonctionne correctement

#### 5.Installation de Dovecot :

#### 

#### 

#### Installe Dovecot, un serveur de messagerie IMAP, pour permettre aux utilisateurs de récupérer leurs courriels

#### ->tester le Dovecot

#### 

#### 

#### 5.Démarrage de Dovecot :

#### “sudo systemctl start dovecot.socket”

#### Démarre le service Dovecot pour permettre l'accès aux boîtes aux lettres via le protocole IMAP

#### 6.Création d'un nouvel utilisateur :

#### “sudo useradd -m -d /home/user1 user1”

#### Crée deux nouveaux utilisateurs sur le système Ubuntu avec un répertoire personnel pour accéder à ses compte de messagerie.

#### 

#### 

#### 7.Installation de Thunderbird :

#### 

#### Installe Thunderbird, un client de messagerie, pour permettre aux utilisateurs d'accéder à leurs courriels via le protocole IMAP.

#### 

#### Envoyer un Message de user2 to user1 :

#### 

#### 