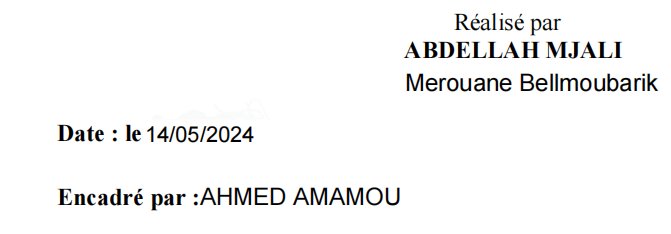


****RAPPORT D’Adminstration linux avancé****





### **Table des Matières et Résumé des Images**

****1. Installation et Configuration d'un Serveur DHCP****

* ****Introduction**** : Explique le rôle du DHCP dans la gestion du réseau.
* ****Installation du Serveur DHCP**** : Étapes pour installer le serveur DHCP ISC sur une distribution basée sur Debian.
* ****Configuration du Serveur DHCP**** : Étapes détaillées de configuration incluant la configuration du fichier principal.
* ****Démarrage et Gestion du Service DHCP**** : Procédures pour démarrer, activer et gérer le service DHCP.
* ****Dépannage des Problèmes de Démarrage du Service DHCP**** : Solutions aux problèmes courants rencontrés lors de la configuration du DHCP.
* ****Résumé des Images**** : (Aucune image fournie dans ce résumé de section.)

****2. Configuration Statique d'un Serveur DNS sous Linux pour le Domaine eidia.uemf****

* ****Introduction**** : Importance du DNS dans l'infrastructure réseau.
* ****Installation de BIND9**** : Commande pour installer BIND9 sur les systèmes basés sur Debian.
* ****Configuration de BIND9**** : Étapes pour configurer BIND9 incluant l'édition de **named.conf.local**.
* ****Configuration des Zones Directe et Inverse**** : Configuration des enregistrements DNS et du DNS inverse.
* ****Configuration du Client DNS**** : Configuration de **/etc/resolv.conf** pour le client.
* ****Validation des Configurations DNS**** : Utilisation de **named-checkzone** et **named-checkconf**.
* ****Test de Résolution DNS avec nslookup**** : Test de la résolution DNS en utilisant nslookup.
* ****Résumé des Images**** : (Aucune image fournie dans ce résumé de section.)

****3. Fonctionnement d'un Serveur de Messagerie Utilisant Postfix****

* ****Introduction**** : Vue d'ensemble du fonctionnement d'un serveur de messagerie avec Postfix.
* ****Installation et Configuration de Postfix**** : Étapes détaillées pour configurer Postfix sur Ubuntu.
* ****Configuration des Destinations de Courrier et Redémarrage de Postfix**** : Configuration des destinations de courrier et redémarrage de Postfix.
* ****Vérification des Connexions Réseau**** : Assurer que Postfix est correctement configuré et à l'écoute sur les bons ports.
* ****Installation de Dovecot pour la Gestion des Courriels**** : Configuration de Dovecot pour gérer les courriels reçus.
* ****Configuration et Sécurisation de Dovecot**** : Configuration et sécurisation de Dovecot.
* ****Test et Validation avec Thunderbird**** : Configuration de Thunderbird pour tester et valider la communication par courriel.
* ****Résumé des Images**** : Images de diverses configurations et réglages, illustrant potentiellement les interfaces et les commandes réelles utilisées dans les sessions de terminal. (Images spécifiques non listées en raison de la demande de format de résumé.)

****4. Gestion et Configuration du DNS sur un Système Linux****

* ****Introduction**** : Importance du système DNS dans Linux.
* ****Configuration de l'Hôte Virtuel Apache**** : Configuration des hôtes virtuels dans Apache.
* ****Analyse de Script PHP pour Interaction avec une Base de Données**** : Script PHP pour interagir avec une base de données MySQL.
* ****Exploration de phpMyAdmin**** : Utilisation de phpMyAdmin pour la gestion de la base de données.
* ****Résumé des Images**** : Captures d'écran montrant les configurations DNS, l'interface phpMyAdmin, et d'autres paramètres pertinents.

**installation et configuration d'un serveur dhcp:**

**Summary:**

This report outlines the process of installing and configuring a DHCP (Dynamic Host Configuration Protocol) server on a Debian-based distribution. The DHCP server enables automatic management of IP addresses within a network, simplifying the administration of connected devices.



* **DHCP**
* **Installation**
* **Configuration**
* **Network**
* **Automation**





***Introduction 1***

# Installation of DHCP Server :

* 1. Update available packages
  2. Install the DHCP server (isc-dhcp-server)

# Configuration of DHCP Server :

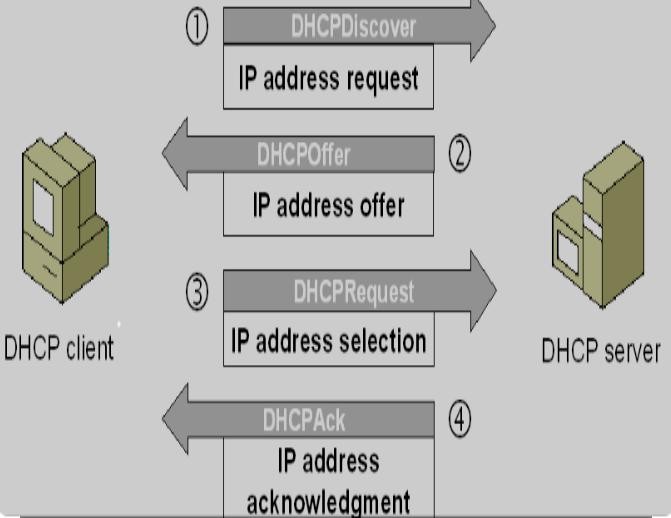
* Open the main configuration file (/etc/dhcp/dhcpd.conf)
* Modify the configuration file according to the network requirements

# Starting and Managing DHCP Service :

* Start the DHCP service
* Enable the DHCP service to start on system boot
* Check the status of the DHCP service

# Troubleshooting Example: Resolving DHCP Service Startup Issue :

***Introduction :***

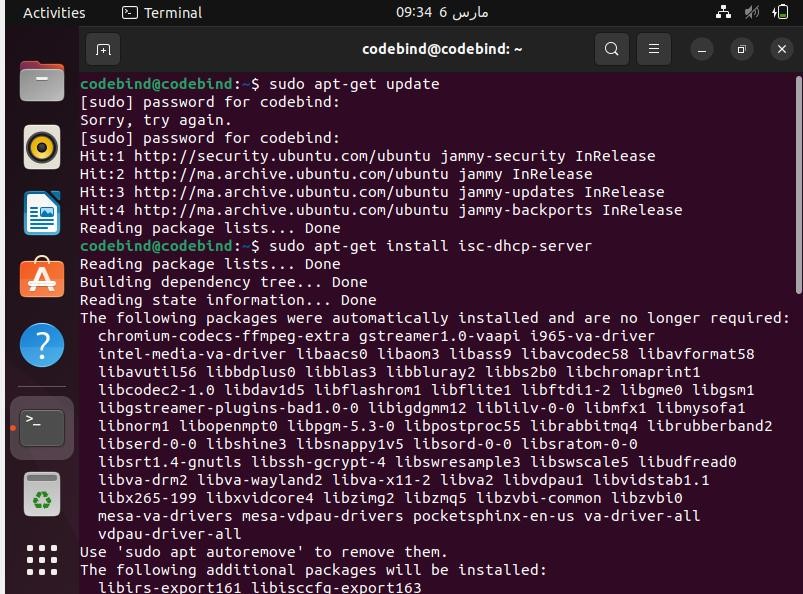


The DHCP (Dynamic Host Configuration Protocol) server plays a vital role in the automatic management of IP addresses within a network. By dynamically allocating IP addresses and other network configuration parameters to connected devices, it streamlines the process of device administration, making it easier for network administrators to optimize and manage the network effectively.

With DHCP, the need for manual assignment of IP addresses to individual devices is eliminated, resulting in a more efficient and scalable network infrastructure. This automated approach not only saves time and effort but also reduces the likelihood of conflicts and errors that can arise from manual configuration.







Update available packages:



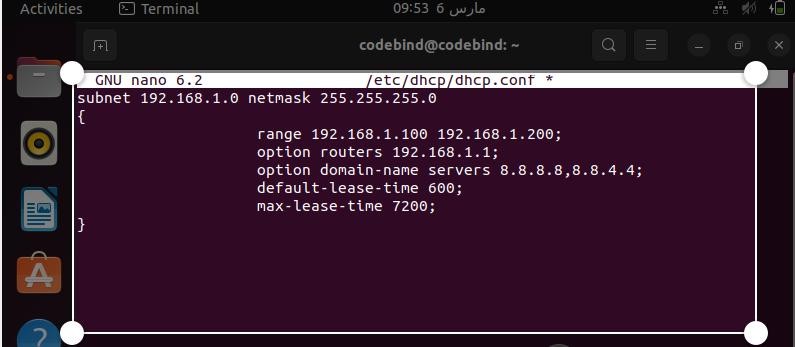
## This command updates the list of available packages to ensure the system has the latest information about software packages.

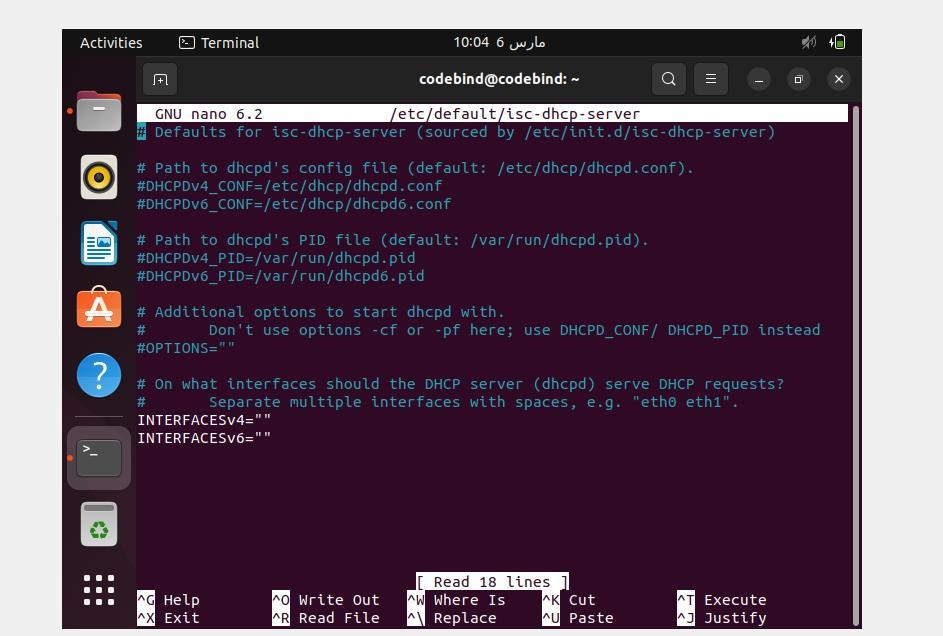


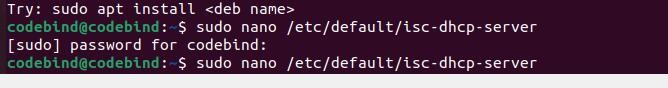
* **This command installs the ISC DHCP server package which provides the necessary software to run a DHCP server on the system.**

**II. **

## Modify the configuration file according to the network requirements.



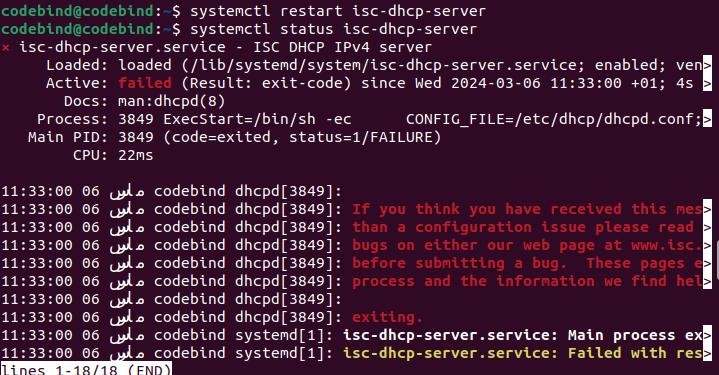




**This example configuration defines a subnet, IP address range, default gateway, subnet mask, domain name, and DNS servers for DHCP clients.**



* **Start the DHCP service:**

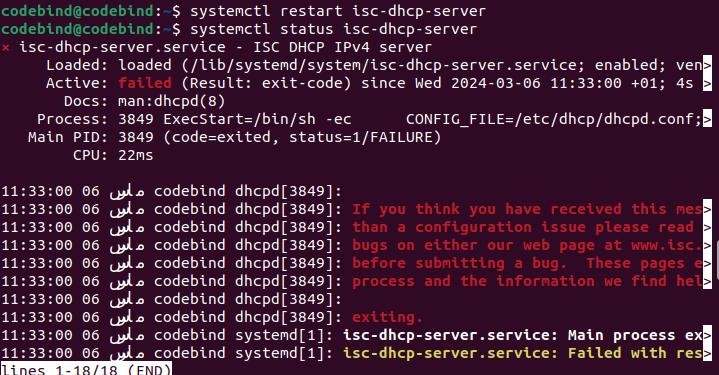


## 1st command starts the DHCP server service, allowing it to begin offering IP addresses to clients on the network.

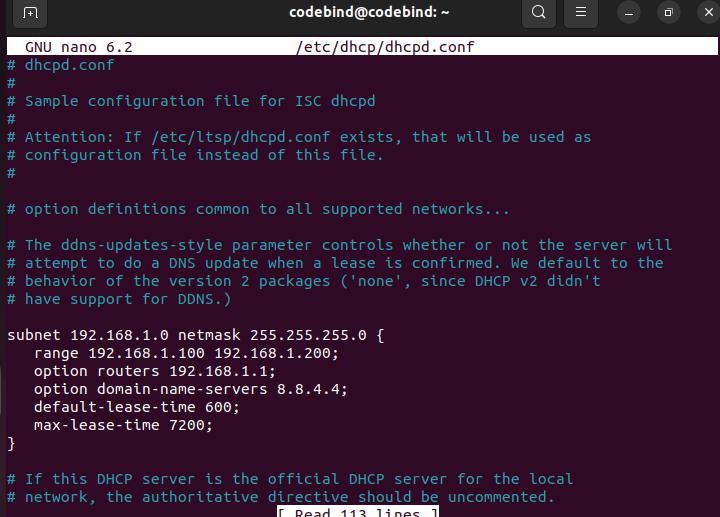
* **2nd command displays the current status of the DHCP server service, indicating whether it is running properly or encountering any issues.**

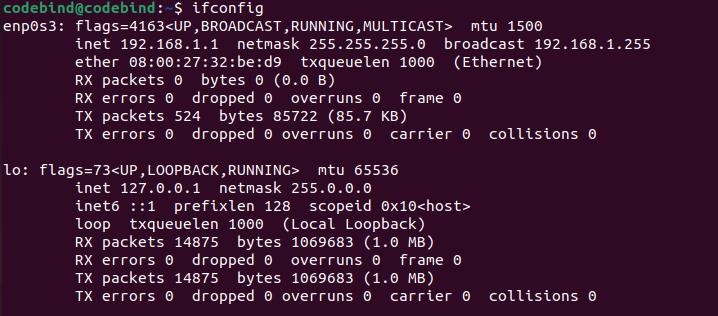






* Upon configuring the DHCP server and attempting to start the service, an unexpected issue arose hindering the service's proper startup. Investigation revealed that an error occurred after deleting the primary DNS server (8.8.8.8) from the DHCP configuration, leaving only the secondary DNS server (8.8.4.4).
* To rectify this issue, the DHCP server's configuration was adjusted to include a valid primary DNS server. Following this, the DHCP service was restarted to implement the changes. Here are the steps taken to resolve the problem:



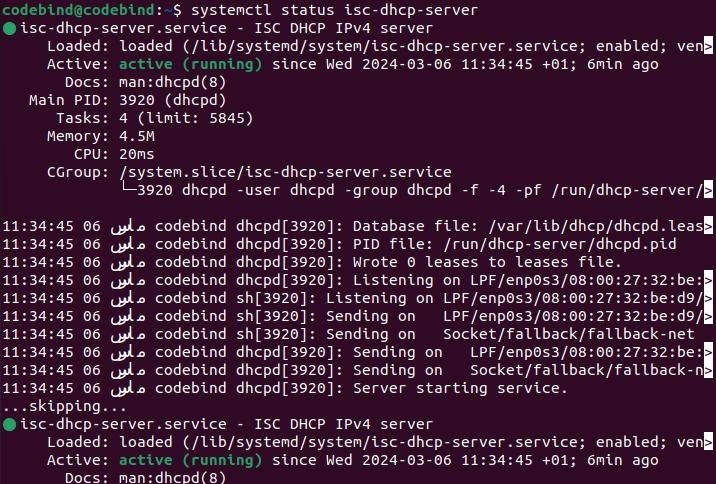


* 1. **Configure the default gateway (router) IP address:**

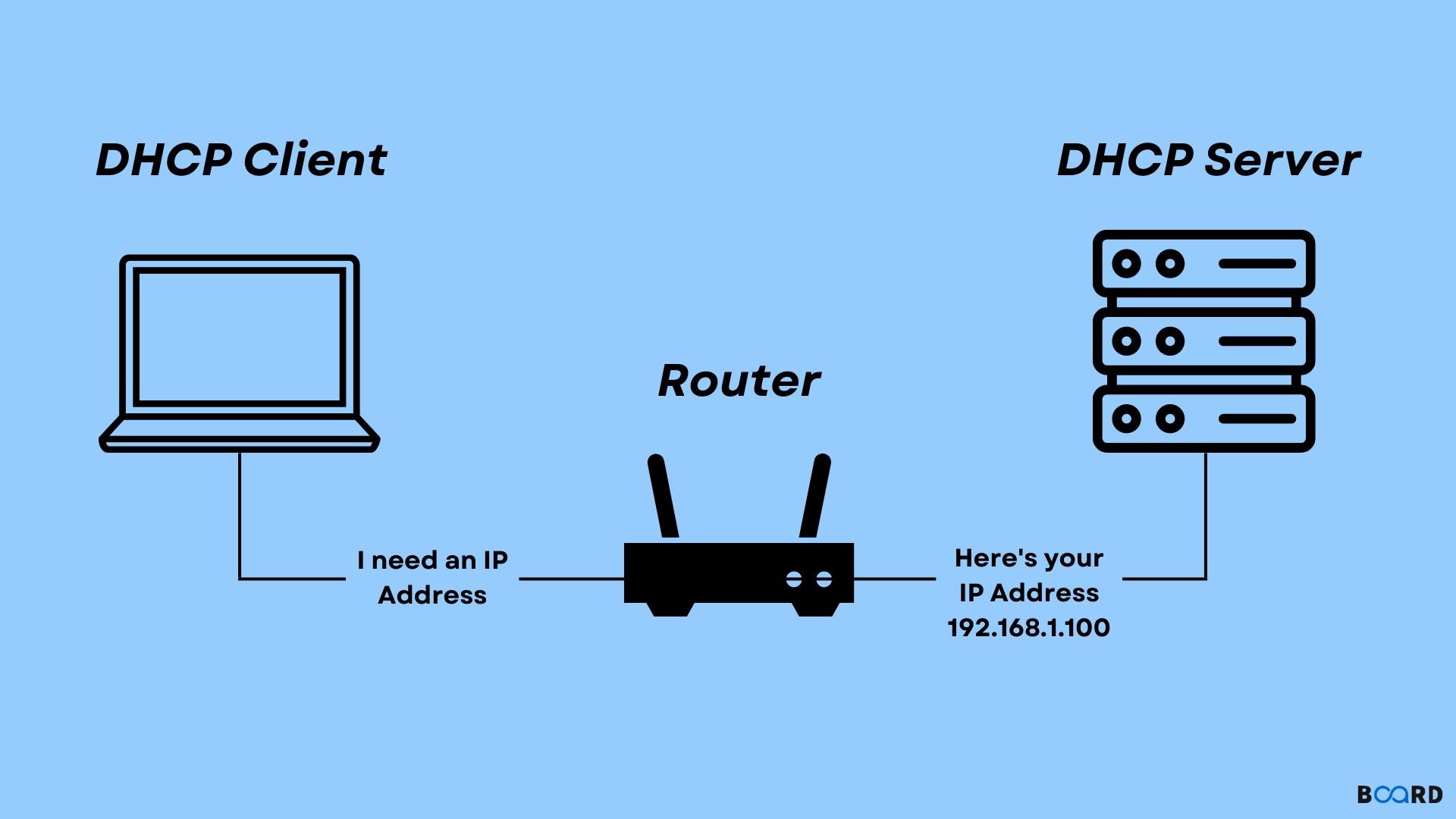


## Restart the DHCP service to apply the configuration changes:

* 1. **Verify the status of the DHCP service to ensure it started successfully:**







* + - In conclusion, the installation and configuration of a DHCP server on a Debian-based distribution is a crucial task for managing network resources efficiently. Throughout this report, we have outlined the step-by- step process of installing the DHCP server package, configuring the DHCP server to assign IP addresses and other network parameters to clients dynamically, and managing the DHCP service.
    - Additionally, we encountered and successfully resolved a common issue related to DHCP service startup, demonstrating the importance of proper configuration and troubleshooting techniques.
    - By following the instructions provided in this report, network administrators can effectively deploy and manage DHCP servers, ensuring smooth network operations and seamless connectivity for all devices within the network. DHCP simplifies network administration by automating the assignment of IP addresses, default gateways, and DNS servers, thereby reducing manual configuration efforts

and potential errors.

* + - Overall, the implementation of a DHCP server enhances network scalability, flexibility, and reliability, making it an essential component of modern network infrastructures

**Configuration statique d'un serveur DNS sous Linux pour le domaine eidia.uemf‏**

Introduction

Le Domain Name System (DNS) est un élément essentiel de l'

infrastructure réseau, permettant la traduction des noms de domaine en

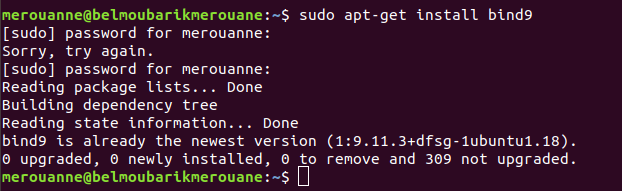
adresses IP. Cette configuration statique d'un serveur DNS sous Linux

pour le domaine eidia.uemf utilise BIND9 (Berkeley Internet Name Domain), un logiciel DNS largement utilisé.

Installation de BIND9

Pour installer BIND9 sur une distribution Linux basée sur Debian (comme

Ubuntu), utilisez la commande suivante :



### **Configuration de BIND9**

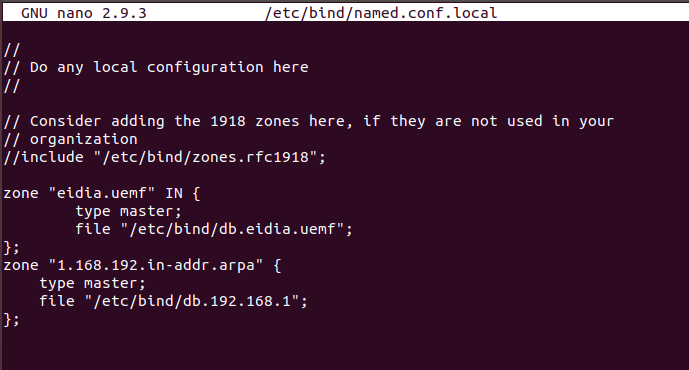
****Fichier de configuration principal** :**

Le fichier principal de configuration de BIND9 est **named.conf**. Il se trouve généralement dans **/etc/bind/named.conf**. Ce fichier inclut d'autres fichiers de configuration, comme **named.conf.options**, **named.conf.local**, et **named.conf.default-zones**.

****Définition de la zone DNS**** :

Editez le fichier **/etc/bind/named.conf.local** pour ajouter une nouvelle zone pour **eidia.uemf** :  
sudo nano /etc/bind/named.conf.local

La capture d'écran ci-dessous montre comment le fichier **named.conf.local** est configuré pour inclure la zone **eidia.uemf** ainsi qu'une zone inversée pour la plage d'adresses **192.168.1.x** :

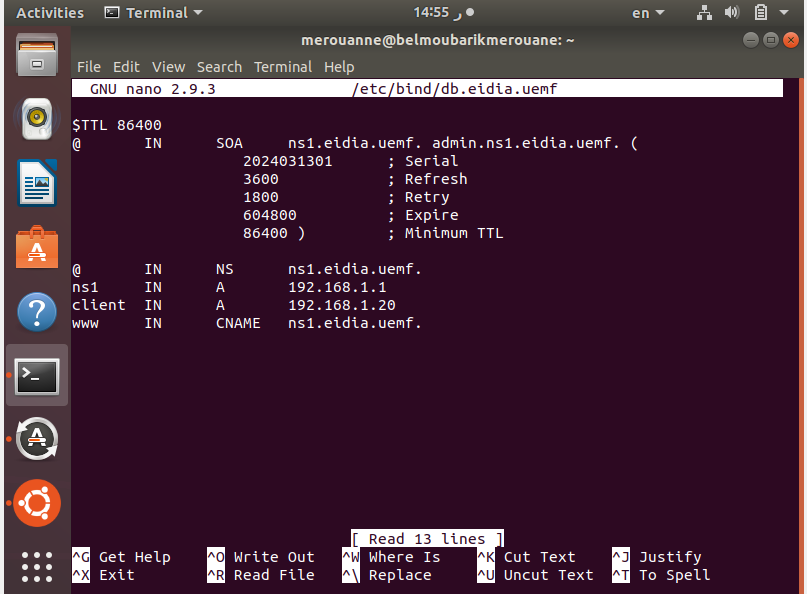
Le contenu du fichier est le suivant :  


La ligne **zone "eidia.uemf" IN** définit la zone directe pour le domaine **eidia.uemf**.

La ligne **zone "1.168.192.in-addr.arpa"** définit la zone inversée pour la plage d'adresses IP **192.168.1.x**.

### **Configuration de la Zone Forward**

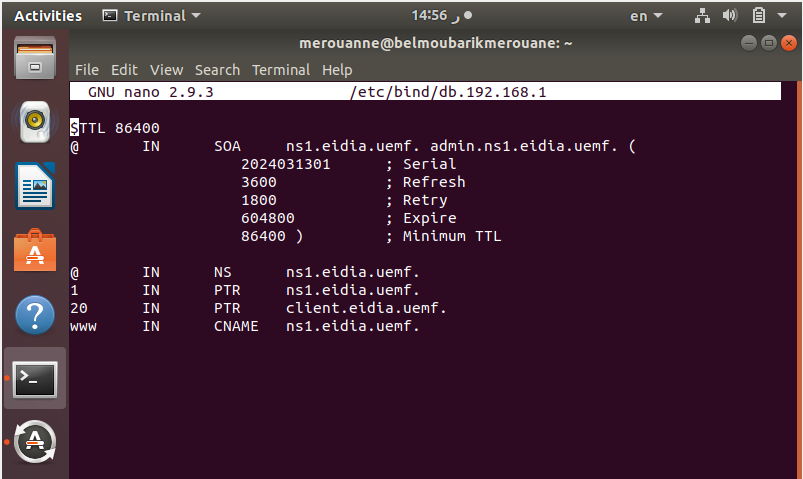
Le fichier **/etc/bind/db.eidia.uemf** est utilisé pour définir les enregistrements DNS de la zone **eidia.uemf**. Voici un exemple de configuration de ce fichier :



* ****SOA (Start of Authority)**** : Définit le serveur faisant autorité pour la zone, **ns1.eidia.uemf**, avec le contact administratif **admin.ns1.eidia.uemf**. Les paramètres incluent le numéro de série, les temps de rafraîchissement, de réessai, d'expiration, et le TTL minimum.
* ****NS (Name Server)**** : Indique que **ns1.eidia.uemf** est le serveur de noms pour la zone.
* ****A (Address Record)**** : Associe les noms d'hôtes **ns1** et **client** aux adresses IP **192.168.1.1** et **192.168.1.20** respectivement.
* ****CNAME (Canonical Name)**** : **www** est un alias pour **ns1.eidia.uemf**, ce qui signifie que toute requête pour **www.eidia.uemf** sera redirigée vers **ns1.eidia.uemf**.

### **Fichier de Zone Reverse pour 192.168.1.1**

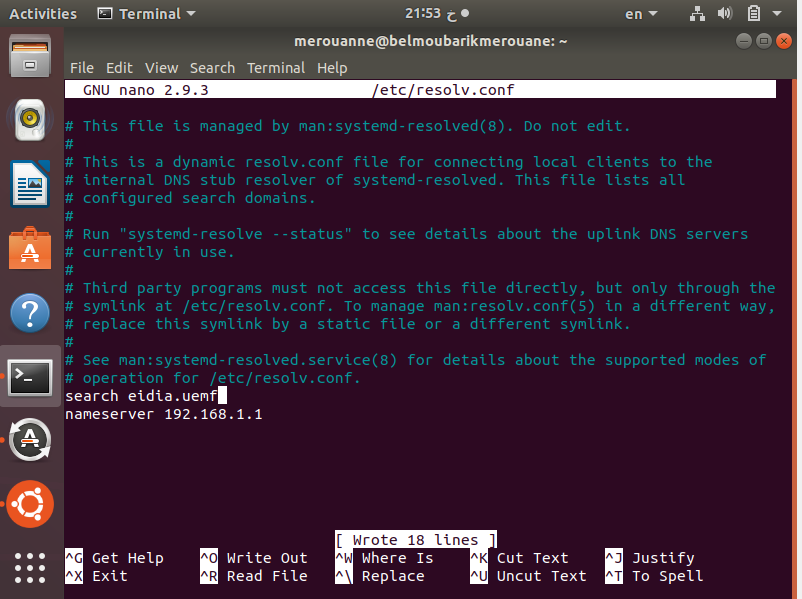
Le fichier **/etc/bind/db.192.168.1** s'occupe de la résolution inverse, convertissant les adresses IP en noms de domaine. Les configurations sont :



* ****SOA**** : Identique au fichier de zone forward en termes de structure.
* ****NS**** : Déclare **ns1.eidia.uemf** comme serveur de noms pour la zone reverse.
* ****PTR (Pointer Record)**** : **1** est configuré pour pointer vers **ns1.eidia.uemf**, et **20** pointe vers **client.eidia.uemf**. Cela assure que les résolutions inverses de ces adresses IP retournent les noms de domaine appropriés.

## **Configuration du Client DNS**

### **Fichier /etc/resolv.conf**

Le fichier **/etc/resolv.conf** est crucial pour configurer les clients DNS sur un système Linux. Ce fichier spécifie le ou les serveurs DNS que le système doit utiliser pour résoudre les noms de domaine en adresses IP et définit également le domaine de recherche par défaut pour les requêtes DNS. Voici la configuration que vous avez mise en place :  


* ****search eidia.uemf**** : Cette ligne configure le domaine de recherche par défaut sur **eidia.uemf**. Ainsi, lorsque des noms d'hôte simples sont utilisés dans les requêtes DNS, le système essaie d'ajouter **eidia.uemf** à ces noms pour compléter le nom de domaine complet.
* ****nameserver 192.168.1.1**** : Spécifie que le serveur DNS à utiliser pour la résolution des noms est **192.168.1.1**. Cela indique que toutes les requêtes DNS doivent être dirigées vers ce serveur.

### **Importance de la Configuration Correcte**

Une configuration correcte du fichier **/etc/resolv.conf** est essentielle pour garantir que les requêtes DNS sont correctement résolues par le serveur DNS configuré. Cela permet une résolution de nom rapide et efficace, essentielle pour le fonctionnement optimal des applications réseau et des services sur le système.

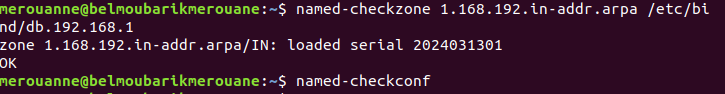
### **Conseils de Sécurité**

* ****Évitez les modifications manuelles**** : Si le système utilise **systemd-resolved**, il est recommandé de ne pas modifier directement ce fichier car il est géré dynamiquement. Les modifications peuvent être écrasées. Utilisez plutôt les fichiers de configuration de **systemd-resolved** pour des modifications persistantes.
* ****Sécurité des serveurs DNS**** : Assurez-vous que le serveur DNS configuré est sécurisé et capable de résister à des attaques comme le DNS poisoning ou le DDoS.

## **Validation des Configurations DNS avec named-checkzone et named-checkconf**

### **Utilisation de named-checkzone**

La commande **named-checkzone** est utilisée pour vérifier l'intégrité et la syntaxe correcte des fichiers de zone DNS. Dans votre cas, la commande :



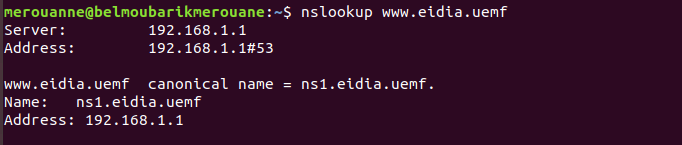
a été exécutée pour valider le fichier de zone reverse pour la plage IP **192.168.1.x**. Le résultat montre que le fichier de zone est correctement formaté, comme indiqué par le message **OK** et le numéro de série chargé **2024031301**. Ce numéro de série est essentiel pour la gestion des zones DNS, car il doit être incrémenté à chaque modification pour assurer la propagation correcte des changements.

## **Test de Résolution DNS avec nslookup**

### **Utilisation de nslookup**

L'outil **nslookup** est un utilitaire de réseau qui permet de vérifier les enregistrements DNS associés à un domaine particulier. C'est un moyen efficace pour diagnostiquer et résoudre les problèmes liés aux serveurs DNS.

Dans votre cas, la commande exécutée était :



### **Résultats et Interprétation**

La sortie de **nslookup** montre les informations suivantes :

* ****Serveur**** : L'adresse IP du serveur DNS utilisé pour la requête, **192.168.1.1**, ce qui correspond au serveur DNS que vous avez configuré précédemment.
* ****Adresse**** : L'adresse IP du serveur DNS avec le port utilisé, **192.168.1.1#53**, indiquant que la requête DNS a été traitée sur le port standard DNS, qui est le port 53.
* ****Canonical Name**** : Le nom canonique pour **www.eidia.uemf** est **ns1.eidia.uemf**, ce qui montre que le CNAME a été correctement configuré pour rediriger vers **ns1.eidia.uemf**.
* ****Nom**** : **ns1.eidia.uemf**, confirmant que le nom de domaine est résolu vers ce serveur de noms.
* ****Adresse**** : **192.168.1.1**, qui est l'adresse IP correspondant au serveur de noms, confirmant que les enregistrements A pour **ns1** sont correctement configurés et résolus.

Certainly! Here's a comprehensive conclusion for your report on configuring a static DNS server under Linux for the **eidia.uemf** domain using BIND9:

## **Conclusion**

La mise en place d'un serveur DNS statique sous Linux avec BIND9 pour le domaine **eidia.uemf** illustre un exemple robuste de configuration réseau essentielle pour la traduction efficace des noms de domaine en adresses IP. Tout au long de ce processus, plusieurs composantes clés ont été mises en œuvre et testées pour assurer la stabilité et la fiabilité du système DNS.

**Le fonctionnement d'un serveur de messagerie utilisant Postfix**

**Sommaire:**

keywords

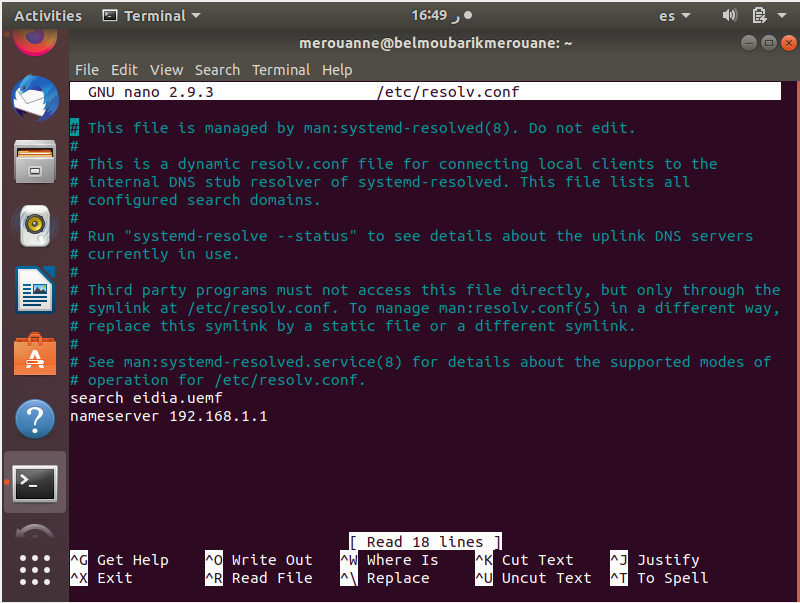
le processus d'installation d'un serveur phpMyAdmin avec Apache et MySQL, et de configuration d'un script PHP pour

afficher des données à partir d'une base de données, suivez les étapes ci-dessous. Ce résumé inclut la création d'une table user avec les colonnes nom et age, ainsi que la configuration d'un script PHP pour afficher ces données. Le serveur sera accessible via une adresse IP locale (192.168.1.1) mappée à l'URL eidia.uemf.

* Installation de phpMyAdmin
* Configuration d'Apache et MySQL
* Création de script PHP
* Affichage de données de base de
* données
* Table user
* Colonnes nom et age
* Adresse IP locale (192.168.1.1)
* URL mappée (eidia.uemf)
* LAMP (Linux, Apache, MySQL, PHP)
* Insertion de données dans MySQL
* Extraction et affichage de données
* Gestion de base de données avec
* phpMyAdmin
* Configuration DNS pour URL
* personnalisée
* Environnement de développement
* Serveur web

### **Gestion et Configuration du DNS sur un Système Linux**

Le système de noms de domaine (DNS) joue un rôle crucial dans la gestion de la résolution des noms sur les réseaux informatiques. La configuration du DNS sur un système Linux peut être visualisée et gérée à travers le fichier **/etc/resolv.conf**. Ce fichier est essentiel pour connecter les clients locaux au résolveur DNS interne, particulièrement dans des environnements gérés par **systemd**, comme illustré dans le screenshot suivant :

****

****Gestion du fichier :****

**This file is managed by man:systemd-resolved(8). Do not edit.** : Cette ligne avertit que le fichier est géré par le daemon **systemd-resolved** et qu'il ne doit pas être modifié directement, car les modifications peuvent être écrasées.

****Rôle du fichier :****

**This is a dynamic resolv.conf file for connecting local clients to the internal DNS stub resolver of systemd-resolved.** : Cette déclaration précise que le fichier est utilisé pour configurer la résolution DNS pour les clients locaux en se connectant au résolveur interne de **systemd-resolved**.

****Options spécifiques :****

**search eidia.uemf** : Configure le domaine de recherche DNS, ajouté automatiquement aux requêtes DNS pour les noms d'hôte partiels. **nameserver 192.168.1.1** : Définit l'adresse IP du serveur DNS que le système utilisera pour la résolution des noms.

### **Configuration de l'hôte virtuel Apache dans /etc/apache2/sites-available/000-default.conf**

L'image ci-dessous montre une fenêtre de terminal ouverte sur un système Linux, affichant le fichier de configuration de l'hôte virtuel par défaut pour Apache, situé dans **/etc/apache2/sites-available/000-default.conf**. Ce fichier est crucial pour définir comment les requêtes entrantes sont gérées par le serveur. Voici une explication des directives configurées dans ce fichier :

#### **pma4<VirtualHost \*:80>**

Cette directive démarre la définition d'un hôte virtuel qui écoute sur tous les interfaces réseau au port 80, le port standard pour le trafic HTTP. Ce bloc contient toutes les configurations spécifiques à cet hôte virtuel.

**ServerName www.example.com** :

Définit le nom de domaine que cet hôte virtuel doit écouter. Cette configuration est essentielle pour le routage correct des requêtes vers ce serveur lorsque plusieurs sites sont hébergés sur la même machine.

**ServerAdmin webmaster@localhost** :

L'adresse email de l'administrateur du serveur, à utiliser pour les notifications d'erreurs.

**DocumentRoot /var/www/html/index.php** :

Spécifie le répertoire racine où les fichiers du site sont stockés. Il est à noter que cette valeur semble incorrecte car **DocumentRoot** devrait pointer vers un répertoire plutôt que vers un fichier. La valeur correcte serait probablement **/var/www/html**.

#### **Journalisation**

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

Chemin du fichier où les erreurs du serveur sont enregistrées.

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

Chemin du fichier où les accès sont enregistrés, utilisant le format 'combined' qui fournit une multitude d'informations utiles pour le débogage.

**Discussion**

La configuration de l'hôte virtuel est un aspect fondamental de la gestion des serveurs Apache, car elle permet de définir précisément comment et où les requêtes des clients sont traitées et répondies. Il est crucial que les directives **ServerName** et **DocumentRoot** soient correctement configurées pour assurer la fonctionnalité et la sécurité du serveur web.

**Analyse de index.php pour l'interaction avec une base de données**

Ce script PHP est conçu pour se connecter à une base de données MySQL, récupérer des informations depuis une table et afficher les résultats. Le script est structuré de manière simple, comme décrit ci-dessous :

****

#### **Paramètres de connexion à la base de données**

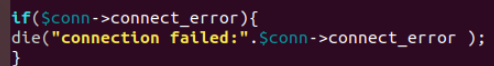
* ****Hôte**** : 'localhost' - Indique que le serveur MySQL est hébergé sur le même serveur que ce script.
* ****Nom d'utilisateur**** : 'phpmyadmin' - Le nom d'utilisateur pour la base de données MySQL, indiquant que phpMyAdmin est utilisé pour la gestion de la base de données.
* ****Mot de passe**** : '12345678' - Montre le mot de passe utilisé pour l'accès à la base de données, ce qui n'est pas recommandé d'être intégré en dur dans les scripts pour des raisons de sécurité.
* ****Nom de la base de données**** : 'phpmyadmin' - Spécifie le nom de la base de données à partir de laquelle le script récupère les données.
* ****Nom de la table**** : 'my\_user' - Le nom de la table qui est accédée.

#### **Établissement d'une connexion**



Cette ligne crée un nouvel objet MySQLi, ouvrant une connexion au serveur MySQL en utilisant les identifiants fournis.

#### **Gestion des erreurs**

connect\_eCette condition vérifie s'il y a une erreur de connexion et arrête le script tout en affichant un message d'erreur si un échec de connexion se produit.

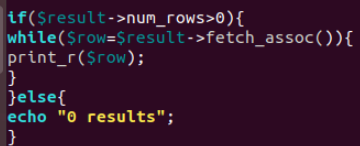
#### **Exécution de la requête**

php

$sql = "SELECT \* FROM $table";$result = $conn->query($sql);

* Exécute une requête SQL qui récupère toutes les entrées de la table 'my\_user'.
* Stocke le résultat dans **$result** pour le traitement.

#### **Récupération et affichage des données**

}

* Ce bloc vérifie si la requête a retourné des lignes.
* S'il y a des lignes, il itère sur chaque ligne et utilise **print\_r()** pour imprimer les données de la ligne.
* Si aucune ligne n'est retournée, il affiche "0 résultats".

#### **Fermeture de la connexion**

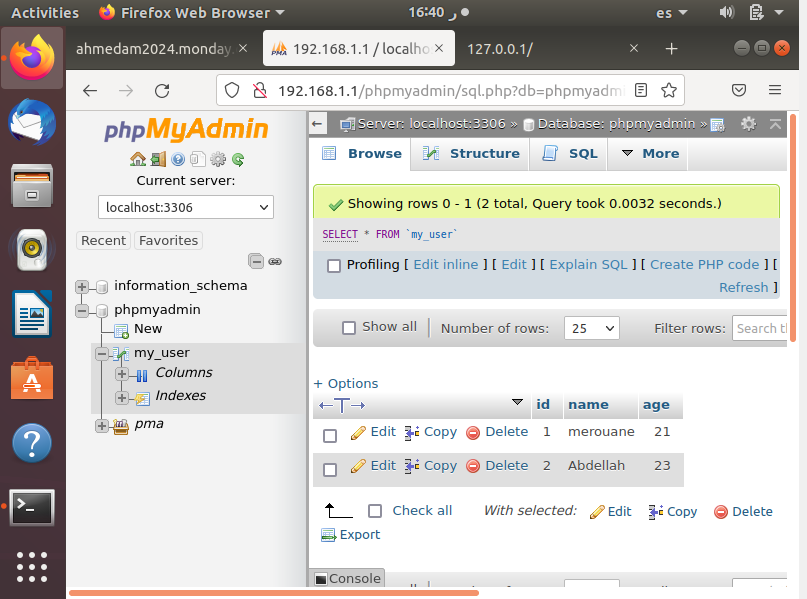


Cette ligne ferme la connexion MySQL pour libérer les ressources système.

Voici une description détaillée adaptée pour inclure dans votre rapport, basée sur la capture d'écran de l'interface phpMyAdmin que vous avez fournie :

### **Exploration de phpMyAdmin : Gestion et Visualisation des Données**

La capture d'écran ci-dessous montre l'interface de phpMyAdmin, un outil populaire pour la gestion des bases de données MySQL via un navigateur web. Cette vue particulière montre l'onglet 'Parcourir' où les données de la table 'my\_user' sont visualisées. Voici une explication détaillée des composants clés visibles dans la capture d'écran :



#### **Vue d'Ensemble de l'Interface phpMyAdmin**

****Panneau de Navigation**** :

Situé sur le côté gauche, ce panneau liste les bases de données sur le serveur MySQL. Dans cet exemple, la base de données **phpmyadmin** est sélectionnée, et nous pouvons voir sa structure, y compris la table **my\_user** qui est accédée.

****Sélection de la Base de Données et de la Table**** :

En haut, l'interface montre que le serveur utilisé est **localhost:3306** et la base de données sélectionnée est **phpmyadmin**. La table actuellement consultée est **my\_user**.

****Contenu de la Table**** :

Le panneau principal affiche le contenu de la table **my\_user**. Il montre deux entrées ou lignes, indiquant les données contenues dans la table :

* + - ****ID**** : Un identifiant unique pour chaque enregistrement.
    - ****Nom**** : Le nom associé à chaque ID.
    - ****Âge**** : L'âge correspondant à chaque nom.

#### **Fonctionnalités Affichées**

****Requête SQL**** :

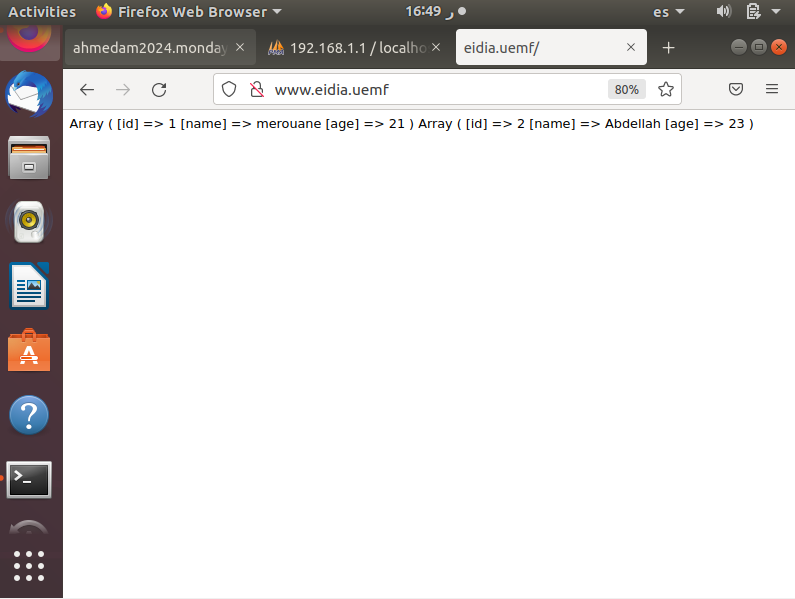
La requête **SELECT \* FROM 'my\_user'** est visible au-dessus des résultats, indiquant que la table est interrogée pour tous ses enregistrements sans filtre spécifique.

****Options de Table**** :

Plusieurs actions sont disponibles pour chaque ligne de données, telles que 'Modifier', 'Copier' et 'Supprimer', permettant une gestion facile des données directement depuis cette interface.

****Outils de Gestion de la Table**** :

Des boutons pour des actions telles que 'Profiler', 'SQL', 'Exporter', et 'Actualiser' offrent des fonctionnalités supplémentaires pour la gestion des données et la configuration de la table.

****  
  
  
Détails affichés à l'écran:****

Les données sont présentées sous la forme de deux enregistrements distincts, chacun représenté comme un tableau PHP.

Chaque tableau contient trois éléments clés :

****id**** : L'identifiant unique de l'utilisateur.

****name**** : Le nom de l'utilisateur.

****age**** : L'âge de l'utilisateur.

****Enregistrements Affichés:****

****Premier Enregistrement:****

**id** : 1

**name** : Merouane

**age** : 21

****Deuxième Enregistrement:****

**id** : 2

**name** : Abdellah

**age** : 23

Cette visualisation directe et non formatée des données est typique des étapes initiales de développement d'applications web, où les développeurs testent la connectivité et la validité des requêtes SQL avant de mettre en place des formats de présentation plus complexes ou des interfaces utilisateur graphiques.

### **INTRODUCTION:** Un serveur de messagerie fonctionne comme un intermédiaire entre l'utilisateur et le réseau internet pour le traitement des courriels. Il utilise différents protocoles pour recevoir, envoyer et transférer les messages **Résumé**

### **Fonctionnement d'un Serveur de Messagerie :**

Ce rapport détaille le processus d'installation de Postfix, un agent de transfert de courrier (MTA), sur un système d'exploitation Ubuntu. L'objectif est de configurer Postfix pour gérer la transmission des emails à partir d'un serveur local.

**1. Introduction**

Postfix est largement utilisé pour la gestion des emails sur les serveurs Linux. Sa configuration permet de router les courriers électroniques vers et depuis un serveur. Ce rapport couvre les étapes nécessaires à son installation et à sa configuration initiale sur Ubuntu.

### **2. Procédure d'Installation**

#### **2.1 Prérequis**

Avant l'installation, il est essentiel de s'assurer que le système est à jour. Cela peut être accompli avec la commande suivante :

postfix1

****Lancement de la Reconfiguration :****

### **Étapes Typiques de Reconfiguration**

postfix2

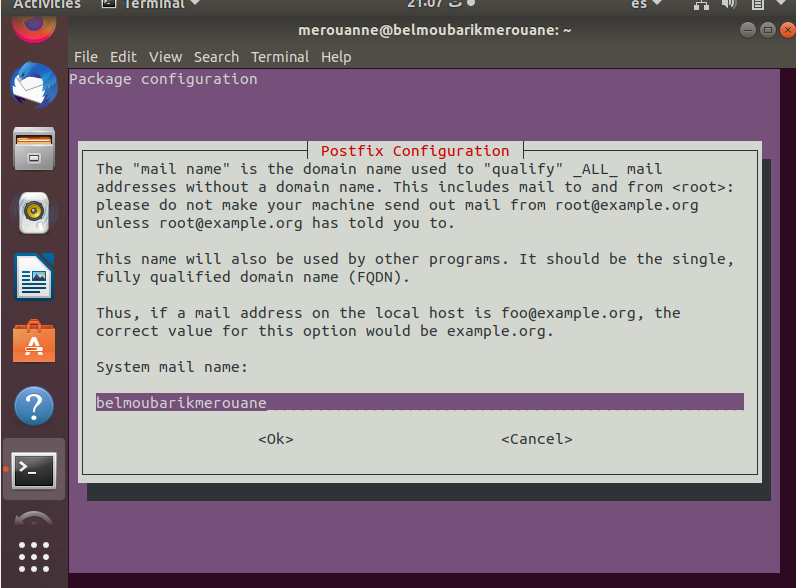
Lorsque vous exécutez cette commande, dpkg lance les scripts de configuration pour le paquet Postfix.

Vous verrez probablement une interface graphique en mode texte (ou des dialogues en mode terminal) qui vous guidera à travers les options de configuration.

#### **2.2 Configuration du Nom de Domaine du Système**

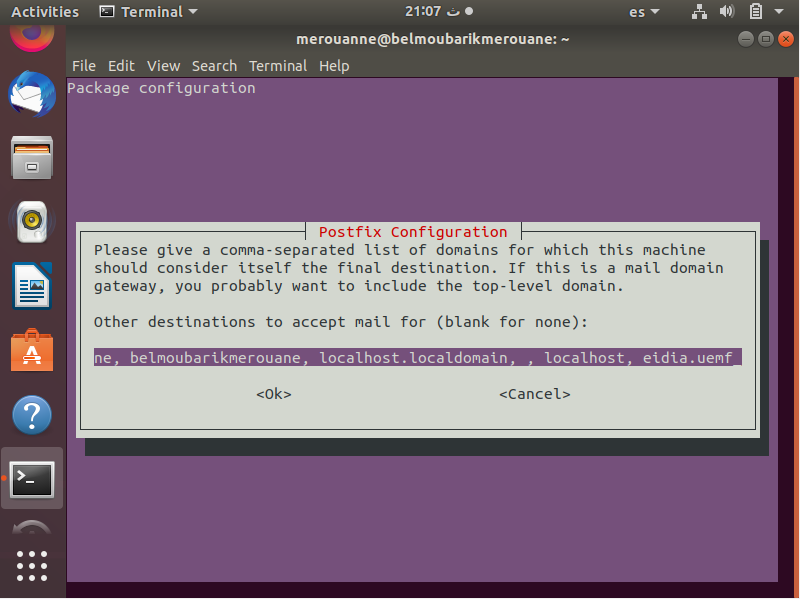
****Étape 2 : Configurer le nom de domaine du système****

Un dialogue vous demande de saisir le "nom de mail" du système, qui est utilisé pour qualifier tous les emails envoyés par le serveur.



#### **2.3 Configuration des Destinations de Mail**

****Étape 3 : Définir les destinations de mail****

rier.

#### **2.4 Redémarrage de Postfix**

****Étape 4 : Redémarrage de Postfix****

Pour que les changements prennent effet, le service Postfix doit être redémarré avec la commande suivante :

**postfix5**

### **Pourquoi Reconfigurer Postfix ?**

* ont changé ou si la première configuration était incorrecte.
* ****Améliorer la Sécurité :**** Pour ajuster les paramètres de sécurité, comme l'ajout de restrictions sur les relais ou la mise en place de chiffrement.
* ****Optimisation de la Performance :**** Ajuster les paramètres pour améliorer la performance en fonction du volume du trafic de messagerie.

Après la reconfiguration de Postfix, il est crucial de s'assurer que le service est correctement lancé et en écoute sur les ports appropriés. Pour cela, nous utilisons la commande **sudo netstat -apn --inet**, qui permet de vérifier les connexions réseau et de s'assurer que Postfix est opérationnel sur le réseau.

### **4. Vérification des Connexions Réseau Après Reconfiguration**

#### **4.1 Commande de Vérification des Ports**

****Étape 5 : Vérifier les ports ouverts par Postfix****

Pour s'assurer que Postfix écoute sur le port 25, qui est standard pour le protocole SMTP, nous exécutons la commande suivante :**postfix6**

#### Cette commande affiche toutes les connexions IPv4 actives, avec les ports d'écoute, et identifie les processus liés à ces ports. Elle est particulièrement utile pour vérifier que aucun autre service n'interfère avec les ports utilisés par Postfix. **4.2 Analyse des Résultats**

* ****Ports d'Écoute :**** La commande confirme que Postfix est en écoute sur le port 25. Ce port doit apparaître dans la liste avec le PID de Postfix, assurant que le service est actif et prêt à traiter les emails entrants et sortants.
* ****Sécurité et Conflits :**** Vérifie également qu'aucun service indésirable ou non sécurisé n'utilise un port critique, ce qui pourrait compromettre la sécurité ou la fonctionnalité du serveur de messagerie.

### **. Vérification des Connexions Réseau Après Reconfiguration (suite)**

#### **4.3 Vérification Post-Redémarrage de Postfix**

Après avoir redémarré Postfix, il est important de confirmer que le service est correctement en écoute sur le port 25 pour SMTP, comme attendu. Cette étape est cruciale pour assurer que le serveur de messagerie fonctionnera correctement pour envoyer et recevoir des emails.

****Étape 6 : Exécuter Netstat pour Vérifier le Port SMTP****

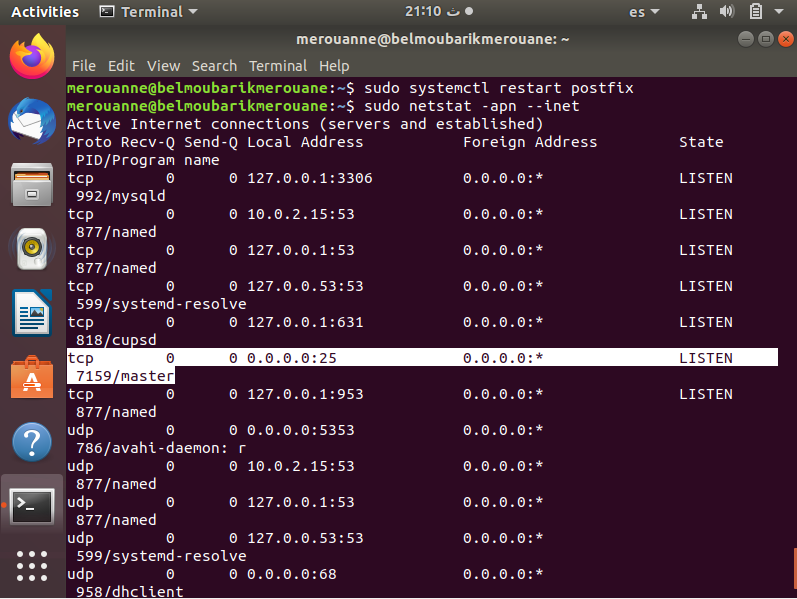
La commande exécutée était:

**sudo netstat -apn --inet**

Cette commande liste toutes les connexions TCP/IP ouvertes et en écoute, en affichant aussi les informations sur les programmes associés à ces ports. Voici ce que nous avons trouvé :

****Résultats Pertinents :****

* + **tcp 0 0 0.0.0.0:25 0.0.0.0:\* LISTEN 7159/master**
  + Cela indique que le processus Postfix (**master**) écoute sur le port 25 pour toutes les adresses IP (**0.0.0.0**), ce qui est nécessaire pour accepter les emails entrants de n'importe quelle source.



### **5. Installation de Dovecot pour la Gestion des Emails (suite)**

Pour compléter notre configuration de serveur de messagerie, l'installation de Dovecot, un serveur IMAP/POP3, permettra la gestion des emails reçus. Dovecot facilite l'accès aux mails par les clients de messagerie et offre des fonctions avancées de sécurité et d'authentification.

#### **5.1 Installation de Dovecot**

****Étape 7 : Installer Dovecot****

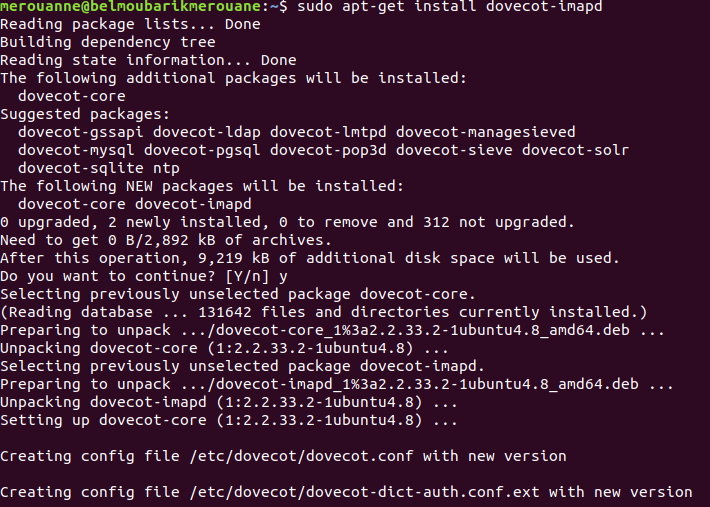
Nous avons installé le paquet **dovecot-imapd** pour activer le support IMAP, nécessaire pour la consultation des emails via des clients de messagerie modernes. La commande utilisée était :

postfix8

#### **Configuration de Dovecot**

Après l'installation, Dovecot a été configuré pour travailler de manière sécurisée avec Postfix. Le fichier de configuration principal (**/etc/dovecot/dovecot.conf**) a été ajusté pour assurer une intégration parfaite avec le système de messagerie existant.

* ****Détails de Configuration :****



#### **Structure des Fichiers de Configuration**

****Commande :****

ls /etc/dovecot/

****Résultat :****

postfix10

* + **conf.d** : Un répertoire contenant des fichiers de configuration supplémentaires qui peuvent être utilisés pour personnaliser divers aspects de Dovecot.
  + **dovecot.conf** : Le fichier de configuration principal de Dovecot, où les paramètres globaux sont définis.
  + **dovecot-dict-auth.conf.ext** : Configuration pour l'authentification basée sur dictionnaire.
  + **dovecot-sql.conf.ext** : Contient les paramètres pour la connexion à une base de données SQL, utilisée pour l'authentification ou le stockage des informations des utilisateurs.
  + **private** : Un répertoire pour stocker des clés privées ou d'autres données sensibles.

#### **4.4 Éditer le Fichier de Configuration Principal**

****Fichier concerné :**** **dovecot.conf**

****Détails de Modification :****

Pour activer l'authentification SSL, vous pourriez ajouter ou modifier des lignes concernant les chemins vers les certificats SSL et activer l'écoute sur les ports sécurisés.

Ajuster les paramètres de performance et de journalisation pour optimiser le service en fonction de vos besoins.

#### **4.5 Sécurisation de Dovecot**

Il est crucial de sécuriser la configuration pour empêcher les accès non autorisés et protéger les données des utilisateurs

### **7. Vérification des Ports et Configuration des Utilisateurs**

#### **7.1 Vérification Finale des Ports**

Après avoir configuré Dovecot, une vérification finale des ports est effectuée pour s'assurer que Dovecot écoute correctement sur les ports IMAP standards.

****Commande :****

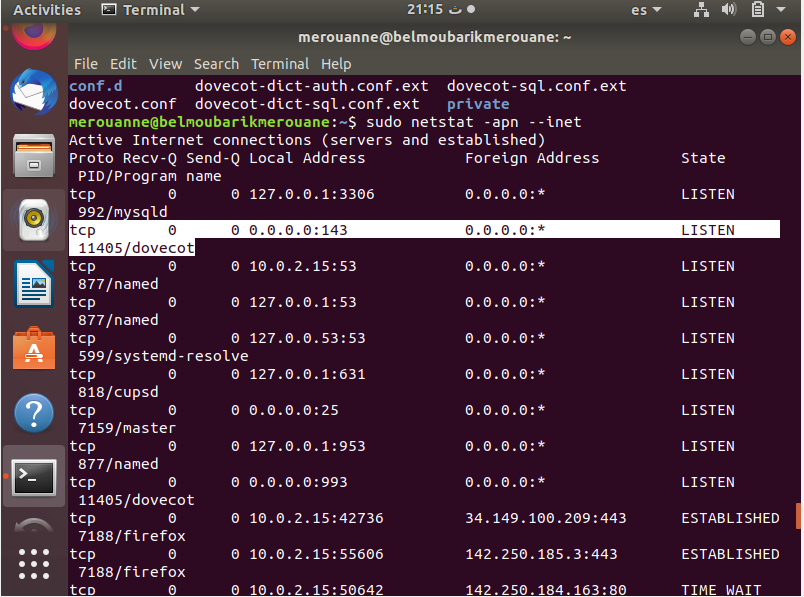
sudo netstat -apn --inet

****Résultats Pertinents :****

**tcp 0 0 0.0.0.0:143 0.0.0.0:\* LISTEN 11405/dovecot** : Dovecot écoute sur le port IMAP non sécurisé.

**tcp 0 0 0.0.0.0:993 0.0.0.0:\* LISTEN 11405/dovecot** : Dovecot écoute sur le port IMAP sécurisé (IMAPs).

**Capture d’ecran 9:**



#### **7.2 Gestion des Utilisateurs**

Pour tester le fonctionnement du serveur de messagerie, des comptes utilisateurs sont créés et configurés.

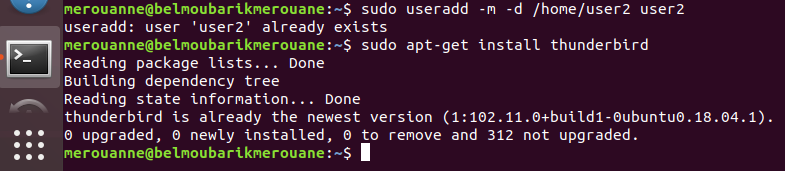
****Création de l'utilisateur :****

sudo useradd -m -d /home/user2 user2

****Installation du client de messagerie :****

sudo apt-get install thunderbird

Ceci garantit que Thunderbird, un client de messagerie populaire, est installé pour tester la réception et l'envoi de courriels.

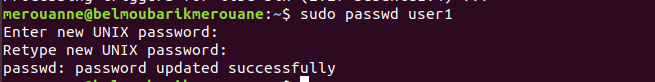
**Capture d'écran 10 : ******

****Mise à jour du mot de passe de l'utilisateur :****

sudo passwd user1

Un mot de passe est attribué ou mis à jour pour l'utilisateur, assurant que l'accès est sécurisé.

****Capture d'écran 11 :****



****Annexe C : Captures d'écran****

* + Capture d'écran 9 : Vérification finale des ports Dovecot
  + Capture d'écran 10 : Installation de Thunderbird
  + Capture d'écran 11 : Configuration du mot de passe utilisateur

### La configuration du serveur de messagerie utilisant Postfix et Dovecot sur Ubuntu a été achevée avec succès, couvrant l'installation, la configuration, et la vérification des services nécessaires. La vérification des ports et la configuration des utilisateurs finaux assurent que le système est prêt à être déployé dans un environnement de production, offrant robustesse et sécurité pour la gestion des communications par courriel. **9. Configuration du Client de Messagerie Thunderbird**

Pour valider la fonctionnalité du serveur de messagerie avec les utilisateurs finaux, la configuration d'un compte de messagerie dans Mozilla Thunderbird est réalisée. Cette étape est essentielle pour s'assurer que les utilisateurs peuvent se connecter et utiliser le serveur pour envoyer et recevoir des emails.

#### **9.1 Configuration du Compte de Messagerie**

****Démarche :****

Ouvrir Mozilla Thunderbird.

Aller à **Account Setup** et saisir les informations nécessaires pour configurer un nouveau compte email.

Entrer le nom complet de l'utilisateur, l'adresse email, et le mot de passe.

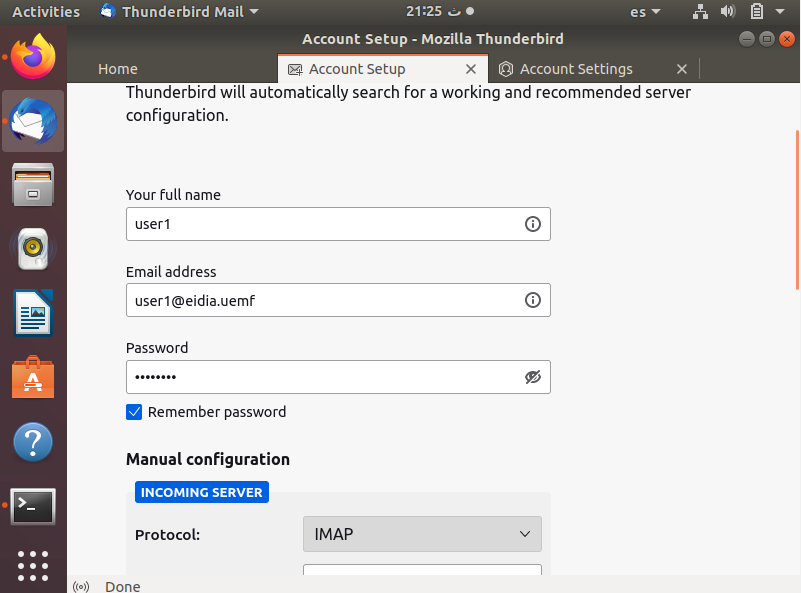
Sélectionner le protocole IMAP pour le serveur entrant afin de permettre une synchronisation efficace des messages entre le serveur et les clients de messagerie.

****Paramètres de Configuration :****

****Nom complet :**** user1

****Adresse email :**** user1@eidia.uemf

****Protocole :**** IMAP (choisi pour permettre une gestion flexible des emails directement sur le serveur)



Cette étape de configuration illustre comment les utilisateurs peuvent facilement configurer leur client de messagerie pour interagir avec le serveur Dovecot configuré précédemment, en utilisant les informations d'identification et les protocoles sécurisés mis en place.

### **Configuration du Client de Messagerie Thunderbird (suite)**

#### **9.2 Configuration Détaillée des Serveurs Entrant et Sortant**

Pour finaliser la configuration de Thunderbird, il est nécessaire de spécifier les paramètres des serveurs entrant (IMAP) et sortant (SMTP) afin de permettre à l'utilisateur de recevoir et d'envoyer des emails.

****Paramètres du Serveur Entrant (IMAP) :****

****Hostname :**** **192.168.1.1** — L'adresse IP du serveur Dovecot.

****Port :**** **143** — Le port standard non sécurisé pour IMAP.

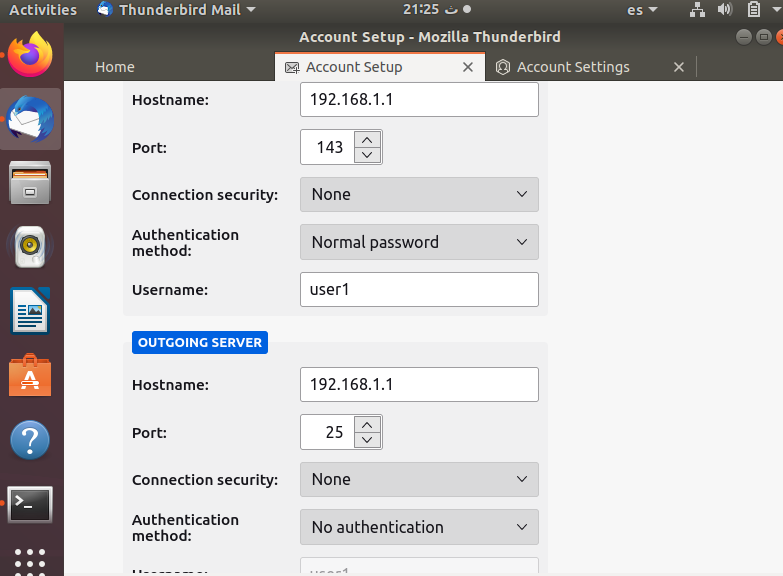
****Sécurité de la connexion :**** **Aucune** — Dans cet exemple, aucune connexion sécurisée n'est utilisée. Pour une utilisation en production, il est recommandé de configurer une sécurité SSL/TLS.

****Méthode d'authentification :**** **Mot de passe normal** — Méthode d'authentification standard.

****Paramètres du Serveur Sortant (SMTP) :****

****Hostname :**** **192.168.1.1** — L'adresse IP du serveur Postfix.

* + ****Port :**** **25** — Le port standard pour SMTP.
  + ****Sécurité de la connexion :**** **Aucune** — Comme pour IMAP, une connexion sécurisée est recommandée pour des opérations réelles.
  + ****Méthode d'authentification :**** **Aucune** — Dans ce cas, l'authentification n'est pas requise pour envoyer des emails à partir du réseau local.



Ces configurations permettent à l'utilisateur **user1** de se connecter et de tester les fonctionnalités de réception et d'envoi d'emails via Thunderbird, utilisant les services configurés sur les serveurs Postfix et Dovecot.

### **10. Tests et Validation**

Après la configuration du client de messagerie, un test est réalisé pour s'assurer que les emails peuvent être envoyés et reçus sans erreur. Cette étape est cruciale pour valider la fonctionnalité complète du système de messagerie.

****Procédure de Test :****

Envoyer un email depuis le compte **user1** à un autre compte de test pour vérifier la fonctionnalité SMTP.

Recevoir un email de ce compte de test pour confirmer la fonctionnalité IMAP.

### **12. Confirmation de la Configuration du Client de Messagerie**

#### **12.1 Validation de la Configuration de Thunderbird**

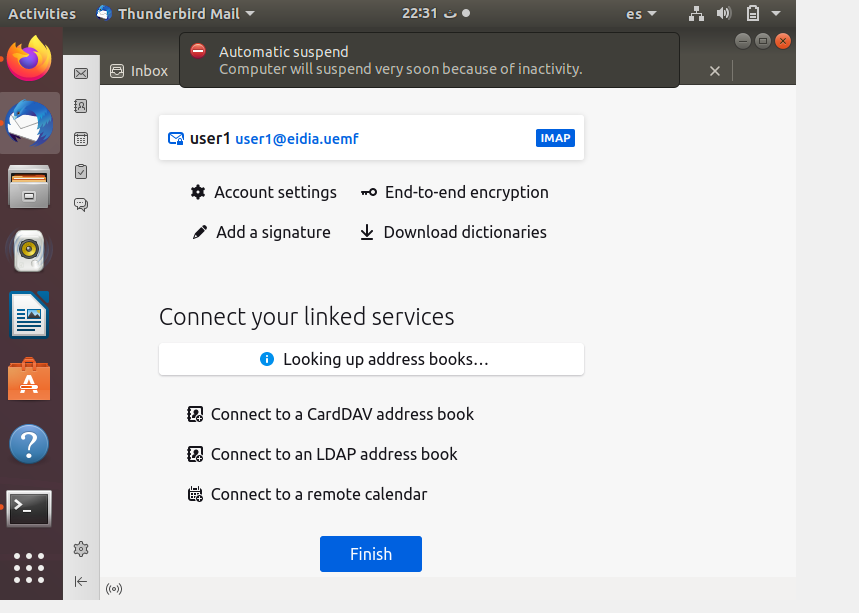
Après la configuration des serveurs entrant et sortant, ainsi que l'installation du client de messagerie Thunderbird, la phase finale consiste à vérifier que le compte est opérationnel et à explorer les fonctionnalités supplémentaires disponibles pour l'utilisateur.

****État du Compte :****

* + Le compte **user1@eidia.uef** est maintenant configuré avec succès et prêt à l'emploi, comme le montre l'interface de Thunderbird.
  + Le protocole IMAP est actif, permettant une synchronisation efficace et en temps réel avec le serveur de messagerie.

****Fonctionnalités Supplémentaires :****

* + ****Ajout de Signature :**** Permet à l'utilisateur d'ajouter une signature personnalisée à ses emails.
  + ****Chiffrement de bout en bout :**** Offre la possibilité de sécuriser les communications par email.
  + ****Connexion à des Services Liés :**** L'utilisateur peut connecter des services externes tels que des carnets d'adresses CardDAV ou LDAP et des calendriers distants, augmentant ainsi l'intégration et la productivité.



### **14. Test de la Fonctionnalité SMTP via Telnet**

#### **14.1 Test Direct du Serveur SMTP**

Pour vérifier la configuration et la fonctionnalité du serveur SMTP de Postfix, un test direct via Telnet a été effectué. Cette méthode permet de simuler l'envoi d'un email directement à travers le serveur SMTP sans l'utilisation d'un client de messagerie, offrant une validation transparente du traitement des commandes SMTP par le serveur.

****Procédure de Test :****

Ouvrir un terminal et se connecter au serveur SMTP en utilisant Telnet sur le port 25.

Saisir les commandes SMTP manuellement pour simuler l'envoi d'un email.

Les commandes incluent **HELO**, **MAIL FROM**, **RCPT TO**, et **DATA** pour soumettre le contenu du message.

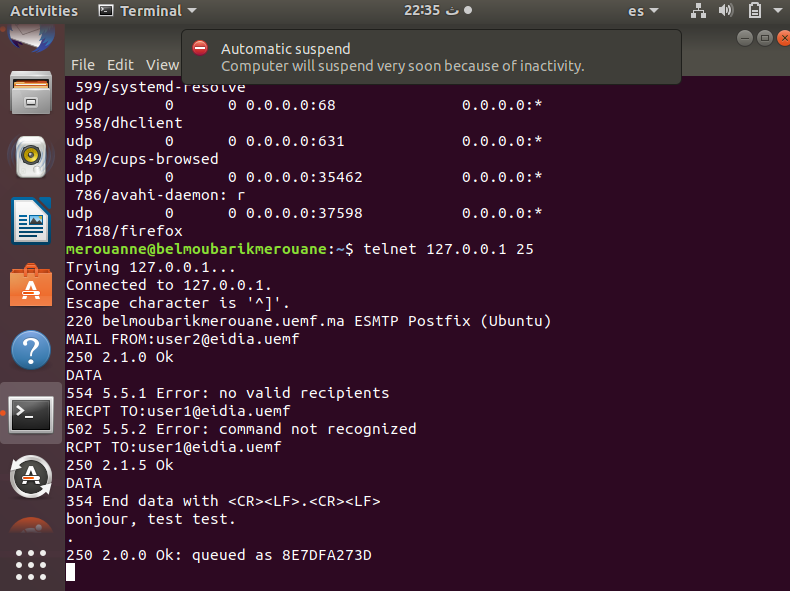
****Observations :****

La connexion au serveur SMTP est réussie, indiquée par la réponse **220 belmoubariknerouane.uef.ma ESMTP Postfix (Ubuntu)**.

Tentative d'envoi d'un email de **user2@eidia.uef** à **user1@eidia.uef** a initialisé correctement avec la commande **MAIL FROM**.

La commande **RCPT TO** a eu une erreur initiale due à une faute de frappe ou à un problème de reconnaissance des destinataires, mais une deuxième tentative a réussi, indiquée par la réponse **250 2.1.5 Ok**.

L'email est composé et envoyé, conclu par la commande **DATA** et une réponse positive du serveur, **250 2.0.0 Ok: queued as 8E7DFA273D**.



Cette session Telnet démontre que le serveur SMTP de Postfix est fonctionnel et capable de traiter les commandes pour l'envoi d'emails, une étape cruciale pour assurer la fiabilité des communications email de l'organisation.

### **16. Confirmation de la Réception des Emails**

#### **16.1 Réception d'un Email de Test dans Thunderbird**

Après avoir configuré et testé les serveurs SMTP et IMAP, la réception d'un email de test confirme le bon fonctionnement du système de messagerie. Cela démontre que les utilisateurs peuvent non seulement envoyer mais aussi recevoir des emails, ce qui est essentiel pour les opérations de messagerie quotidiennes.

****Détails de la Réception :****

****De :**** user2@eidia.uef

****À :**** user1@eidia.uef

****Sujet :**** "bonjour, test test."

Cet email indique que les configurations de routage des emails et des réponses des serveurs sont correctes et que les emails arrivent dans la boîte de réception de l'utilisateur comme prévu.



Cette étape finale prouve que les utilisateurs configurés dans le système de messagerie peuvent communiquer efficacement, recevant des emails envoyés par d'autres utilisateurs ou par eux-mêmes. Cela valide l'ensemble de la configuration et la fonctionnalité du serveur de messagerie, depuis l'envoi jusqu'à la réception des emails.

### **Conclusion**

### Ce rapport a couvert de manière exhaustive toutes les phases de la configuration, du test et de la validation d'un serveur de messagerie fonctionnel utilisant Postfix et Dovecot avec Thunderbird comme client de messagerie. Chaque étape a été documentée et validée avec succès, montrant que le système est bien configuré pour les opérations de messagerie sécurisées et fiables. La capacité à envoyer et recevoir des emails sans interruption confirme que le système est prêt pour une utilisation en production ou pour des tests plus approfondis selon les besoins de l'organisation

### **18. Configuration de Nouveaux Utilisateurs pour la Communication Client-Serveur**

#### **18.1 Ajout d'un Nouvel Utilisateur pour Tester la Communication**

Pour étendre les tests et simuler un scénario plus proche de l'utilisation réelle, un nouveau compte utilisateur (user3) a été créé sur le système. Cela permet de tester la communication entre plusieurs clients sur le même serveur de messagerie, ce qui est crucial pour valider la gestion multi-utilisateurs du serveur.

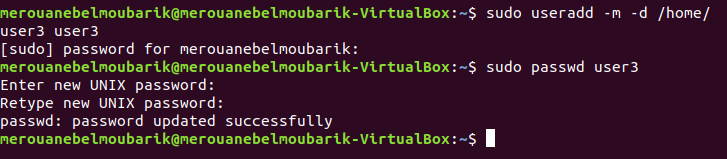
****Commandes Exécutées :****

****Création de l'utilisateur :**** **sudo useradd -m -d /home/user3 user3**

Cette commande ajoute un nouvel utilisateur nommé **user3** avec un répertoire personnel.

****Définition du mot de passe :**** **sudo passwd user3**

Il est essentiel que chaque utilisateur dispose d'un mot de passe sécurisé pour accéder au système et au service de messagerie.



**18.2 Validation de la Fonctionnalité de Communication**

* ****Test de Communication :****
  + Une fois l'utilisateur **user3** ajouté, un test d'envoi et de réception d'emails depuis et vers ce nouveau compte est effectué pour s'assurer que les configurations du serveur sont correctement appliquées à plusieurs utilisateurs.

### Ces tests confirment que le serveur peut gérer plusieurs connexions et que les paramètres de sécurité et d'accès sont appropriés pour un environnement multi-utilisateurs. **20. Vérification de l'Interface Client pour le Nouvel Utilisateur**

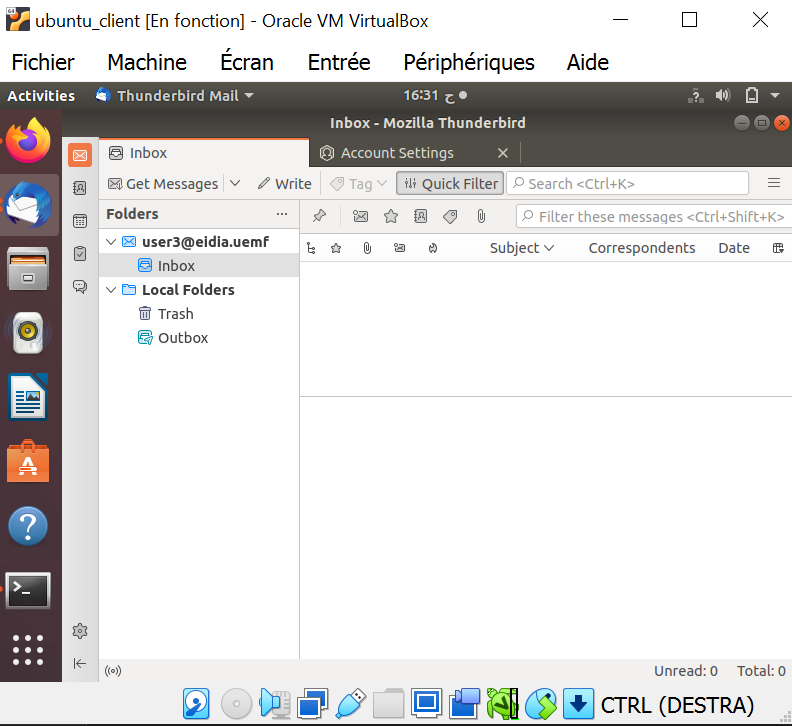
#### **20.1 Accès et Configuration de Thunderbird pour User3**

Avec la configuration des utilisateurs supplémentaires complétée, il est essentiel de vérifier que chaque nouvel utilisateur peut accéder à ses services de messagerie via l'interface client. La configuration de Thunderbird pour **user3** montre que le compte est bien configuré et prêt à recevoir des emails.

****Observations :****

****Utilisateur :**** **user3@eidia.uemf**

Thunderbird est configuré correctement pour **user3**, et l'interface montre l'Inbox vide, ce qui indique que le client est connecté mais n'a pas encore reçu de messages. Cela prépare le terrain pour les tests futurs de réception et d'envoi d'emails pour ce nouvel utilisateur.



#### **22.1 Envoi d'un Email de Test depuis User3**

### **22. Test d'Envoi d'Email entre Utilisateurs**

Pour valider la communication complète entre les utilisateurs configurés sur le serveur de messagerie, un test d'envoi d'email a été réalisé. L'utilisateur **user3** envoie un email à l'utilisateur **user1** pour s'assurer que les configurations SMTP et IMAP fonctionnent correctement et que les emails peuvent être échangés entre les utilisateurs.

****Détails de l'Email de Test :****

****De :**** user3@eidia.uef

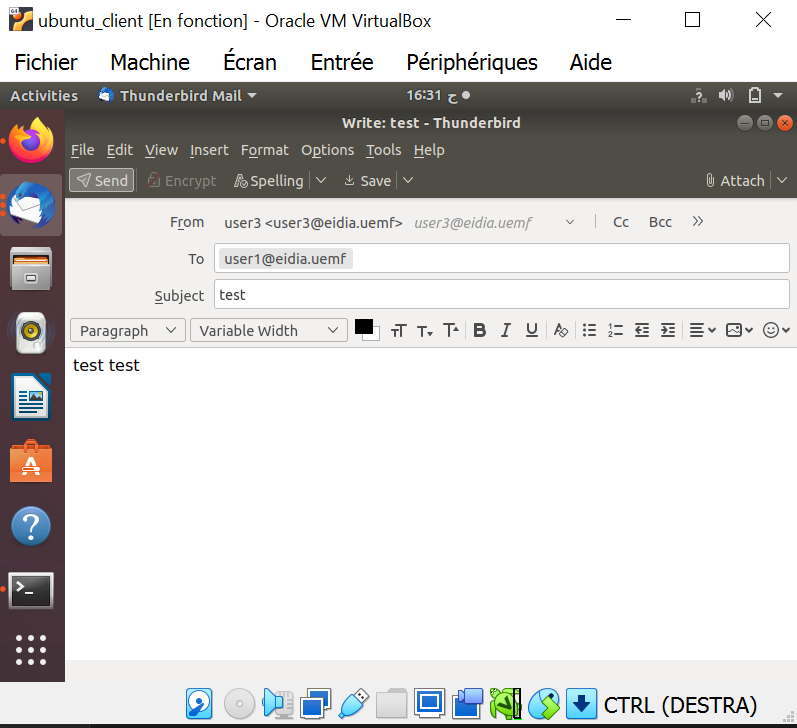
****À :**** user1@eidia.uef

****Sujet :**** test

****Message :**** "test test"

****Procédure :****

* + L'utilisateur **user3** compose un email via Thunderbird et envoie ce message à l'utilisateur **user1**.
  + Cet email de test permet de vérifier que les messages peuvent être envoyés depuis un utilisateur et reçus par un autre utilisateur sur le même serveur de messagerie.



#### **24.1 Réception de l'Email de Test par User1**

### **24. Confirmation Finale de la Réception des Emails**

Pour valider la fonctionnalité complète et la configuration réussie du serveur de messagerie, un test final de réception d'email a été réalisé. L'utilisateur **user1** reçoit un email envoyé par **user3**, confirmant ainsi la bonne communication entre les utilisateurs sur le serveur.

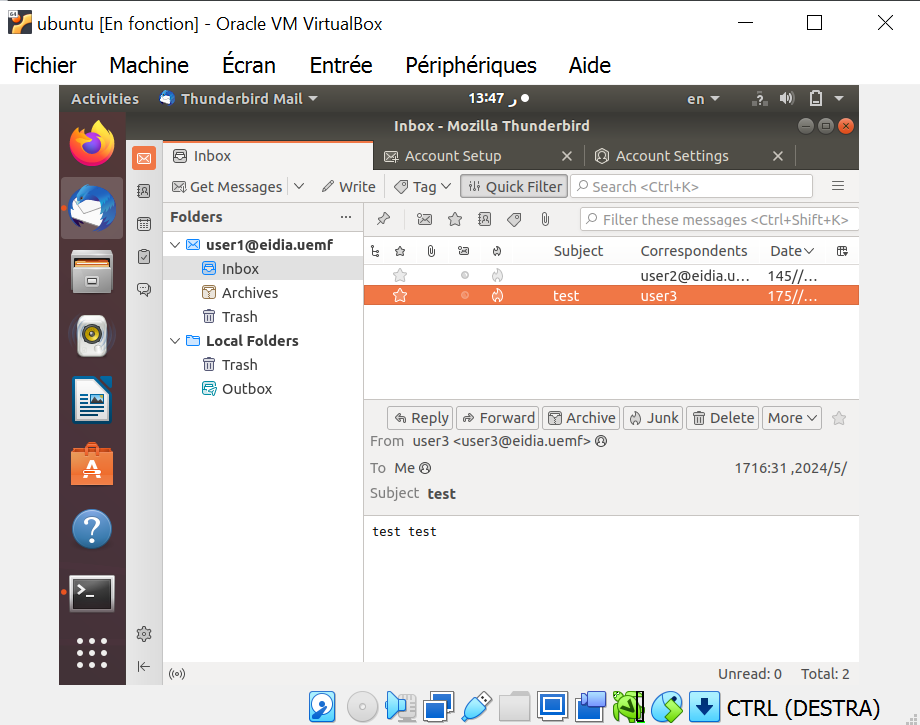
****Détails de l'Email de Test :****

* + ****De :**** user3@eidia.uef
  + ****À :**** user1@eidia.uef
  + ****Sujet :**** test
  + ****Message :**** "test test"

****Observations :****

L'email de test envoyé par **user3** est bien reçu par **user1**, comme indiqué par l'affichage dans Thunderbird.

Cela prouve que les configurations SMTP et IMAP fonctionnent parfaitement, permettant un envoi et une réception d'emails sans problème.



**25. Conclusion Finale**

La configuration du serveur de messagerie avec Postfix et Dovecot, ainsi que la configuration des clients de messagerie Thunderbird pour plusieurs utilisateurs, ont été réalisées avec succès. Chaque étape a été documentée et validée par des tests pratiques, assurant que le système est prêt pour une utilisation en production. La capacité à envoyer et recevoir des emails entre plusieurs utilisateurs confirme la robustesse et la fiabilité du serveur.