

REPUBLIQUE DU SENEGAL



ECOLE SUPÉRIEURE POLYTECHNIQUE / SENELEC



DEPARTEMENT GENIE INFORMATIQUE

Filaire : Informatique

MEMOIRE DE FIN DE CYCLE

Pour l'obtention du :

DIPLÔME SUPERIEUR DE TECHNOLOGIE (DST)

Lieu de stage : **SENELEC** Période stage : **01/07/2023-31/07/2023**

SUJET : Administration d'une base de données PostgreSQL

Présenté par :

Mohamed Mbaye

Maitre de stage

Mr. Oumar Ndiaye

Sous la supervision de :

Dr. Mouhamadou Lamine BA

Année académique : 2022 / 2023

DEDICACES

Nous dédions ce mémoire à :

- ❖ Ma mère Arame THIOUNE et mon père Babacar MBAYE qui peuvent être fières et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans les études. Puisse Dieu faire en sorte que ce travail porte leur fruit ; merci pour les valeurs nobles, l'éducation et le soutien permanent venus de votre part.
- ❖ Nos professeurs de l'ESP qui doivent voir dans ce travail la fierté d'un savoir bien acquis.
- ❖ A toute notre famille, ce travail est le fruit de l'espoir que vous avez de nous voir réussir.
- ❖ A tous mes camarades de classe, mes parrains et filleuls.
- ❖ A mon maître de stage et mon encadreur pour leur soutien et leur engagement tout au long de mon projet.

REMERCIEMENTS

Je rends grâce à DIEU, le Tout Puissant, présent au commencement et à la fin de toute chose.

Mon remerciement.

- Mes sincères remerciements à l'endroit de l'encadreur Docteur Mouhamadou Lamine BA qui a accepté de diriger ce mémoire, et qui est resté disposé et disponible pour apporter sa remarque, ses critiques et suggestions.
- Les encouragements, conseils et soutiens des Professeurs nous ont été d'un grand réconfort tout au long de notre formation académique et de la rédaction de ce mémoire. Qu'ils trouvent ici l'expression de notre profonde gratitude.
- **A mes parents, mes frères et mes sœurs**

J'exprime ma profonde gratitude pour tous les sacrifices consentis, la confiance, ainsi que leur soutien indéfectible. Puisse Dieu le tout puissant exhausser tous vos vœux.

- **À mon maitre de stage, Mr Oumar Ndiaye**, Chef de service production et gestion des données

Merci aussi pour l'encadrement dont j'ai bénéficié durant cette période de stage. Pour votre disponibilité, vos bénéfiques recommandations, grâce aussi à votre confiance, j'ai pu m'accomplir totalement les objectifs que nous nous étions fixé.

- Mes remerciements à l'équipe de la Direction des systèmes d'information qui m'ont bien accueilli et intégré dans leur équipe.

AVANT-PROPOS

L'Ecole Supérieure Polytechnique de Dakar (ESP) est un établissement public de formation professionnelle doté d'une personnalité juridique et d'une autonomie financière. Elle fait partie intégrante de l'Université Cheikh Anta Diop de Dakar (UCAD).

Elle a été créée en 1994 et a pour mission de former, tant sur le plan théorique que pratique, des techniciens supérieurs, des ingénieurs technologues, des ingénieurs de conception, des managers en gestion d'entreprise et des docteurs, de dispenser un enseignement supérieur en vue de préparer aux fonctions d'encadrement dans divers domaines tels que la production, la recherche appliquée, etc.

L'ESP compte (6) départements et dans le cadre de leur formation les étudiants en fin de cycle sont tenus d'effectuer un stage pratique au sein d'une entreprise ou d'un service informatique. Ce stage permet à l'étudiant :

- De renforcer son savoir et surtout d'acquérir un savoir-faire, tout en essayant d'adapter ses connaissances aux cadres de la vie professionnelle.
- De travailler sur un projet de fin d'études et de mener à bien l'élaboration de celui-ci depuis l'étude préalable jusqu'à sa mise en œuvre.

C'est dans ce cadre que nous a été offerte la possibilité d'effectuer un stage de fin d'études au sein de la société **SENELEC** pour une durée de 60 jours.

TABLE DES MATIERES

DEDICACES.....	II
REMERCIEMENTS	III
AVANT-PROPOS	IV
RESUME.....	XII
INTRODUCTION GENERAL.....	1
1 CHAPITRE I : PRESENTATION DU SUJET ET DE LA STRUCTURE D'ACCUEIL	2
1.1 Présentation de la structure d'accueil	2
1.1.1 Définition	2
1.1.2 Activités et organigramme de la SENELEC	2
1.1.3 Présentation de la DSİ.....	3
1.2 Présentation du sujet.....	3
1.2.1 Contexte.....	3
1.2.2 Problématique	4
1.2.3 Objectifs	4
2 CHAPITRE II : ANALYSE DES BESOINS ET MODELISATION.....	5
2.1 Analyse des besoins	5
2.1.1 Besoins fonctionnels	5
2.1.2 Besoins non fonctionnels	5
2.2 Modélisation.....	6
2.2.1 Présentation UML.....	6
2.2.2 Diagrammes de cas d'utilisation	9
2.2.3 Description textuelle des cas d'utilisation	9
2.2.4 Conclusion	13
3 CHAPITRE III : IMPLEMENTATION ET PRESENTATION DES RESULTATS.....	14
3.1 Les outils et technologies utilisés.....	14
3.2 Présentation de PostgreSQL	15
3.2.1 Pourquoi utiliser PostgreSQL ?	15
3.2.2 Schéma de la base de données	16
3.2.3 Installation	16
3.2.4 Configuration.....	19
3.3 La réPLICATION	32
3.3.1 Configuration de la réPLICATION	33

3.3.2	Test de la réPLICATION.....	40
3.3.3	Conclusion	42
CONCLUSION GENERAL.....		43
WEBOGRAPHIE		44

LISTE DES TABLEAUX

Tableau 1 : Configuration réPLICATION	10
Tableau 2 :Gérer Utilisateur.....	11
Tableau 3 : Effectuer une Transaction.....	12
Tableau 4 : Surveiller la réPLICATION.....	13

LISTE DES FIGURES

Figure 1 : LOGO SENELEC	2
Figure 2 : Organigramme de la SENELEC	3
Figure 3 : logo UML.....	6
Figure 4 : évolution UML	6
Figure 5 : les diagrammes UML	8
Figure 6 : Diagramme de cas d'utilisation	9
Figure 7 : diagramme de classe.....	13
Figure 8 : logo Ubuntu	14
Figure 9 : logo Virtual Box	15
Figure 10 : base de donnée 'test'	16
Figure 11 : la table 'stagiaire'.....	16
Figure 12 : logo PostgreSQL.....	15
Figure 13 :Mettre à jour les paquets.....	17
Figure 14 : installer PostgreSQL.....	17
Figure 15 :informations de l'installation de PostgreSQL	18
Figure 16 :Ajout du mot de passe superutilisateur	19
Figure 17 : Modification méthode de connexion.....	20
Figure 18 :Redémarre service PostgreSQL.....	20
Figure 19 : Connexion avec mot de passe.....	21
Figure 20 : Site de téléchargement de PostgreSQL.....	21
Figure 21 : Script à copier	22
Figure 22: exécution du script.....	24
Figure 23 : installation de pgAdmin4	25

Figure 24 : exécution du script 'setup-web.sh'	26
Figure 25 : Saisie du e-mail et mot de passe	27
Figure 26 :Initialisation de pgadmin4.....	28
Figure 27 : installation de pgadmin4.....	29
Figure 28 : fin de l'installation de pgadmin4	30
Figure 29: interface de connexion pgadmin4	31
Figure 30 : interface d'accueil pgadmin4	32
Figure 31 : création du rôle de replication.....	33
Figure 32 : chemin de postgresql.conf.....	34
Figure 33 : activation de la réPLICATION.....	34
Figure 34 : paramètre listen_addresses	35
Figure 35 : chemin de pg_hba.conf.....	35
Figure 36 : configuration de connexion du SE sur le SM.....	36
Figure 37 : redémarrage du service PostgreSQL.....	36
Figure 38 : test de communication entre le SM et le SE.....	37
Figure 39 : sauvegarde du main directory	37
Figure 40 : suppression du main.....	37
Figure 41 : sauvegarde de base du main SM	38
Figure 42 : contenu du nouveau main	39
Figure 43 : statut de PostgreSQL.....	39
Figure 44 : information détaillé de la réPLICATION	40
Figure 45 : Essaie de transaction sur SE	40
Figure 46 : création de base de donnée 'test' sur le SM.....	40
Figure 47 : base de donnee 'test' bien repliqué sur le SE.....	41

Figure 48 :création de la table stagiaire dans la base de donnée 'test' sur le SM 41

Figure 49 : table stagiaire bien répliqué sur le SE 42

SIGLES ET ABREVIATIONS

SE	Serveur esclave
SM	Serveur maître
SENELEC	Société Nationale d'Électricité du Sénégal
DSI	Direction des Systèmes d'Information
UML	Unified Modeling Language

RESUME

Le mémoire se concentre sur l'administration d'une base de données PostgreSQL dans le contexte d'une entreprise SENELEC, plus précisément au sein du service de la Direction des Systèmes d'Information. L'objectif principal de ce projet est d'améliorer la gestion de la réPLICATION des données dans l'entreprise en utilisant PostgreSQL, en établissant des bases solides grâce à une installation adéquate, une configuration, il vise à renforcer la disponibilité et les performances des systèmes d'information de SENELEC, contribuant ainsi à l'efficacité globale de l'entreprise.

Ce mémoire sera particulièrement bénéfique pour la Direction des Systèmes d'Information en ce qui concerne l'installation et la configuration de la réPLICATION dans leurs serveurs, en répartissant la charge sur plusieurs serveurs, en amélioration de la disponibilité et l'efficacité des ressources matérielles et logicielles.

Ce document va décrire les différentes étapes et méthode utilisées pour mener à bien ce projet.

INTRODUCTION GENERAL

La gestion efficace des données est importante pour toute organisation moderne, quelle que soit son secteur d'activité. Dans un monde de plus en plus numérisé, la capacité à stocker, gérer et sécuriser les informations est essentielle pour prendre des décisions, améliorer l'efficacité opérationnelle et rester compétitif sur le marché.

C'est dans ce contexte que la SENELEC m'a proposé un mémoire, qui se penche sur l'administration d'une base de données PostgreSQL, en mettant l'accent sur son installation, sa configuration et la mise en place de la réPLICATION.

Ce mémoire est organisé en plusieurs chapitres qui couvrent chaque aspect de notre projet d'administration de base de données PostgreSQL. Après cette introduction générale, le chapitre suivant sera consacré à la présentation de la structure d'accueil et du sujet. Par la suite, le chapitre 2 sera consacré à l'analyse des besoins et à la modélisation de la base de données. Enfin, nous aborderons l'implémentation et la présentation de notre solution, en mettant en évidence les outils et les technologies utilisés.

1 **CHAPITRE I : PRESENTATION DU SUJET ET DE LA STRUCTURE D'ACCUEIL**

1.1 **Présentation de la structure d'accueil**

1.1.1 Définition

La Société nationale d'électricité du Sénégal (SENELEC) est une société anonyme à capitaux publics majoritaires, concessionnaire de la production, du transport, de la distribution et de la vente de l'énergie électrique au Sénégal. [1]



Figure 1 : LOGO SENELEC

1.1.2 Activités et organigramme de la SENELEC

La SENELEC s'occupe de l'électrification en milieu urbain, qui reste rentable de par la concentration des usagers, tandis que l'électrification rurale est déléguée à l'Agence Sénégalaise de l'Électrification Rurale (ASER). Elle participe aussi au développement de l'énergie solaire photovoltaïque avec des projets comme Senergy 2, financé par GreenWish Partners à Bokhol.

Elle est membre du Système d'échanges d'énergie électrique ouest africain, EEEOA. [2]

Nous avons en dessous l'organigramme de la SENELEC :

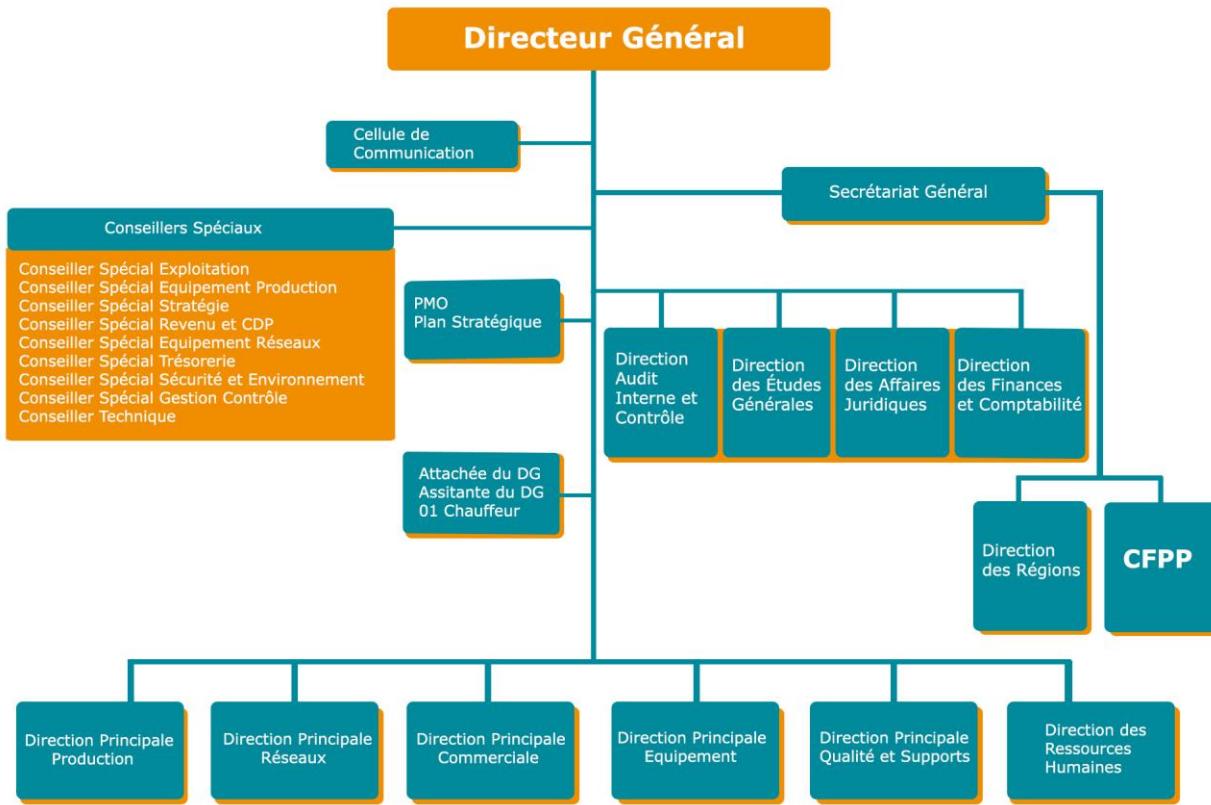


Figure 2 : Organigramme de la SENELEC

1.1.3 Présentation de la DSi

Elle élabore le plan directeur informatique optimal, est responsable de la sécurité informatique et assure la gestion du parc de matériel informatique, la maintenance et l'exploitation du matériel et des logiciels de gestion. Elle assiste les directions dans la mise en place d'applications et d'outils spécialisés. Elle a en charge les projets informatiques, de téléphonie et ceux relatifs au réseau de communication voix et données. Elle assure l'implantation de nouveaux progiciels et la formation des utilisateurs. Elle gère l'infrastructure du réseau de communication voies et données (WAN et LAN, faisceaux hertziens, boucle locale radio), la téléphonie, les fax.^[3]

1.2 Présentation du sujet

Dans cette section, nous allons élaborer une description globale de notre sujet. Pour ce faire, nous commençons par contextualiser notre projet d'étude afin de bien analyser et ressortir la problématique à traiter. Ensuite, nous identifions un certain nombre d'objectifs à atteindre.

1.2.1 Contexte

La SENELEC (Société Nationale d'Électricité du Sénégal) est une entreprise publique chargée de la distribution et de la fourniture d'électricité au Sénégal. Comme de nombreuses organisations modernes, la SENELEC dépend fortement des systèmes d'information pour assurer une gestion

efficace de ses opérations. La Direction des Systèmes d'Information (DSI) joue un rôle essentiel dans la gestion de ces systèmes, notamment en ce qui concerne la gestion de la base de données PostgreSQL, qui stocke et gère des données critiques pour l'entreprise.

1.2.2 Problématique

La problématique centrale de ce projet réside dans la difficulté de configurer PostgreSQL pour répondre aux besoins croissants de l'entreprise en termes de gestion des données, de stockage et de traitement. Comment mettre en place et gérer efficacement la réPLICATION des données dans PostgreSQL au sein de la SENELEC pour assurer la disponibilité continue des données tout en garantissant la cohérence et l'intégrité des données répliquées ?

1.2.3 Objectifs

L'objectif principal de ce projet est d'améliorer la gestion de la réPLICATION des données au sein de la SENELEC à travers PostgreSQL.

Les objectifs spécifiques sont listés ci-après :

- ✓ Etudier PostgreSQL et sa réPLICATION.
- ✓ Installer et configurer PostgreSQL pour une bonne mise en marche.
- ✓ Configurer et tester la réPLICATION.

2 **CHAPITRE II : ANALYSE DES BESOINS ET MODELISATION**

Dans ce chapitre, nous nous plongerons dans l'essence même du projet, en nous efforçant de comprendre les besoins spécifiques qui l'ont engendré et en créant des modèles pour les représenter de manière précise et exploitable.

2.1 Analyse des besoins

L'analyse des besoins est un processus qui permet de définir les attentes et les exigences des utilisateurs ou des clients pour un produit ou un service.

2.1.1 Besoins fonctionnels

Les besoins fonctionnels sont des exigences spécifiques en matière de fonctionnalités que votre projet devra satisfaire pour répondre aux objectifs définis.

- L'administrateur pourra :
 - Gestion des utilisateurs ou groupes : Créer, modifier ou supprimer des utilisateurs ou des groupes.
 - Configuration de la réPLICATION : Autoriser la configuration de la réPLICATION en temps réel pour garantir la synchronisation immédiate des données.
- L'utilisateur pourra :
 - Effectuer une transaction SQL : L'utilisateur peut créer et d'envoyer une transaction au système.
 - Surveillance de la réPLICATION : Fournir une interface de surveillance en temps réel permettant de suivre l'état de la réPLICATION.

2.1.2 Besoins non fonctionnels

Les besoins non fonctionnels sont des exigences qui concernent la manière dont le système doit fonctionner, plutôt que ce que le système doit faire.

- Performance : Les opérations de réPLICATION doivent s'effectuer rapidement pour assurer la synchronisation en temps réel.
- Sécurité :
 - Les données répliquées doivent être sécurisées en utilisant des protocoles de chiffrement robustes.
 - Les mécanismes d'authENTICATION et d'autORISATION doivent être solides pour empêcher l'accès non autorisé.
- Documentation : Une documentation complète et à jour doit être fournie pour faciliter

l'administration et la maintenance du système.

2.2 Modélisation

La modélisation des besoins est une étape cruciale dans le processus de gestion de projet. Elle nous permet de traduire les besoins identifiés lors de l'analyse en des représentations visuelles ou conceptuelles, créant ainsi une base solide sur laquelle le projet peut être développé et géré de manière efficace. Dans ce chapitre nous utiliseront le langage UML pour modéliser notre système informatique.

2.2.1 Présentation UML

Le Langage de Modélisation Unifié, de l'anglais Unified Modeling Language (UML), est un langage de modélisation graphique à base de pictogrammes conçu comme une méthode normalisée de visualisation dans les domaines du développement logiciel et en conception orientée objet.

L'UML est une synthèse de langages de modélisation objet antérieurs : Booch, OMT, OOSE. Principalement issu des travaux de Grady Booch, James Rumbaugh et Ivar Jacobson, UML est à présent un standard adopté par l'Object Management Group (OMG). UML 1.0 a été normalisé en janvier 1997 ; UML 2.0 a été adopté par l'OMG en juillet 2005. La dernière version de la spécification validée par l'OMG est UML 2.5.1 (2017).



Figure 3 : logo UML

2.2.1.1 Evolution du langage

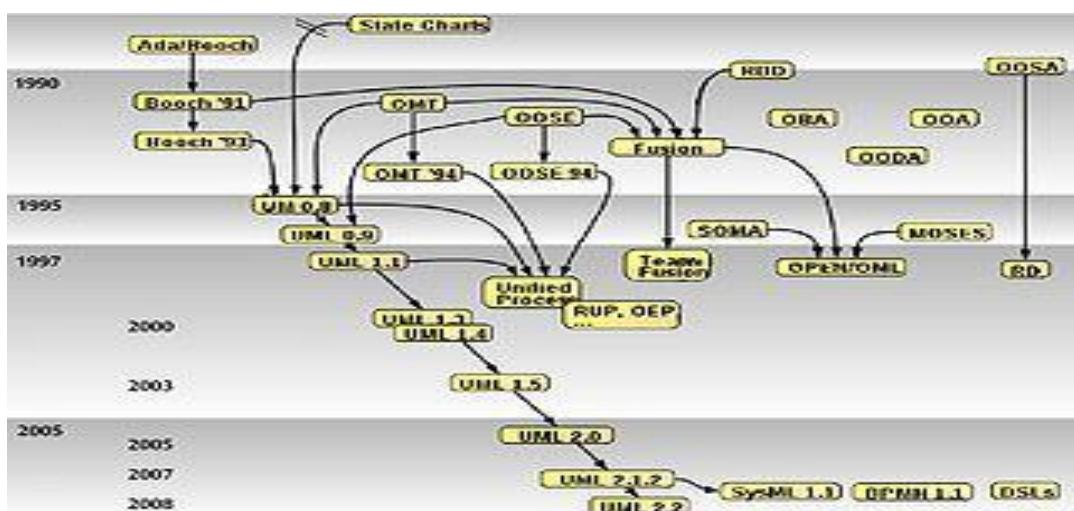


Figure 4 : évolution UML

2.2.1.2 Diagrammes

Les diagrammes sont dépendants hiérarchiquement et se complètent, de façon à permettre la modélisation d'un projet tout au long de son cycle de vie. Il en existe quatorze depuis UML 2.3.

- **Diagrammes de structure ou diagrammes statiques**

Les diagrammes de structure (structure diagrams) ou diagrammes statiques (static diagrams) rassemblent :

- Diagramme de classes (class diagram) : représentation des classes intervenant dans le système.
- Diagramme d'objets (object diagram) : représentation des instances de classes (objets) utilisées dans le système.
- Diagramme de composants (component diagram) : représentation des composants du système d'un point de vue physique, tels qu'ils sont mis en œuvre (fichiers, bibliothèques, bases de données...).
- Diagramme de déploiement (deployment diagram) : représentation des éléments matériels (ordinateurs, périphériques, réseaux, systèmes de stockage...) et la manière dont les composants du système sont répartis sur ces éléments matériels et interagissent entre eux.
- Diagramme des paquets (package diagram) : représentation des dépendances entre les paquets (un paquet étant un conteneur logique permettant de regrouper et d'organiser les éléments dans le modèle UML), c'est-à-dire entre les ensembles de définitions.
- Diagramme de structure composite (composite structure diagram) : représentation sous forme de boîte blanche des relations entre composants d'une classe (depuis UML 2.x).
- Diagramme de profils (profile diagram) : spécialisation et personnalisation pour un domaine particulier d'un méta-modèle de référence d'UML (depuis UML 2.2).

Diagrammes de comportement

Les diagrammes de comportement (behavior diagrams) rassemblent :

- Diagramme des cas d'utilisation (use-case diagram) : représentation des possibilités d'interaction entre le système et les acteurs (intervenants extérieurs au système), c'est-à-dire de toutes les fonctionnalités que doit fournir le système.
- Diagramme états-transitions (state machine diagram) : représentation sous forme de machine à états finis du comportement du système ou de ses composants.

- Diagramme d'activité (activity diagram) : représentation sous forme de flux ou d'enchaînement d'activités du comportement du système ou de ses composants.

Diagrammes d'interaction ou diagrammes dynamiques

Les diagrammes d'interaction (interaction diagrams) ou diagrammes dynamiques (dynamic diagrams) rassemblent :

- Diagramme de séquence (sequence diagram) : représentation de façon séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou de ses acteurs.
 - Diagramme de communication (communication diagram) : représentation de façon simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets (depuis UML 2.x).
 - Diagramme global d'interaction (interaction overview diagram) : représentation des enchaînements possibles entre les scénarios préalablement identifiés sous forme de diagrammes de séquences (variante du diagramme d'activité) (depuis UML 2.x).
 - Diagramme de temps (timing diagram) : représentation des variations d'une donnée au cours du temps (depuis UML 2.3).

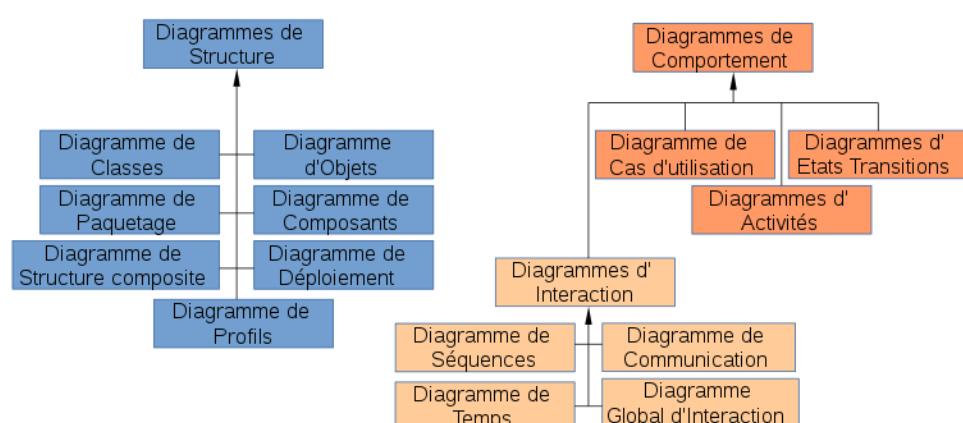


Figure 5 : les diagrammes UML

2.2.2 Diagrammes de cas d'utilisation

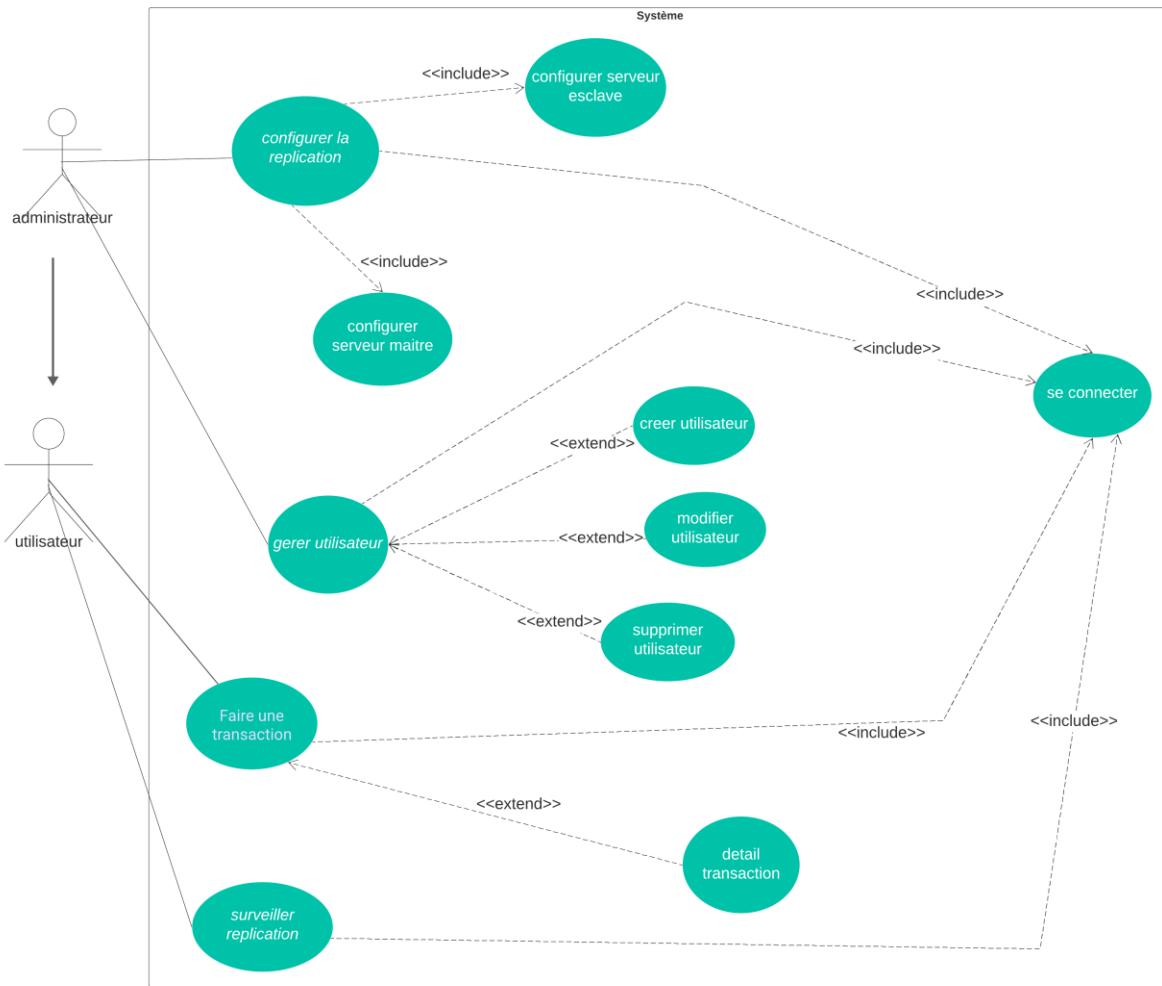


Figure 6 : Diagramme de cas d'utilisation

2.2.3 Description textuelle des cas d'utilisation

Ces tableaux ci-dessous vont décrire les cas d'utilisation suivante :

- Configurer réPLICATION
- Gérer Utilisateur
- Effectuer une Transaction
- Surveiller la réPLICATION

Titre	Configurer réPLICATION
Description/objectif	Ce cas d'utilisation permet à un administrateur du système de configurer la réPLICATION de PostgreSQL en utilisant la console de commande
Acteur	Administrateur
Précondition	<ul style="list-style-type: none"> • L'administrateur est authentifié sur le serveur PostgreSQL • Il existe au moins deux serveurs PostgreSQL : un serveur principal et un serveur de réPLICATION (esclave) déjà configuré
Scénario nominale	<ol style="list-style-type: none"> 1. L'administrateur ouvre une session sur la console de commande du serveur PostgreSQL. 2. À l'aide de commandes SQL ou de scripts, l'administrateur configure la réPLICATION en spécifiant les détails du serveur de réPLICATION, tels que l'adresse IP, le port, les identifiants d'authentification, etc. 3. L'administrateur choisit le mode de réPLICATION souhaité (par exemple, réPLICATION synchrone ou asynchrone) en modifiant les paramètres appropriés dans le fichier de configuration PostgreSQL. 4. L'administrateur valide la configuration en redémarrant le serveur PostgreSQL principal. 5. Le système effectue les étapes nécessaires pour établir la réPLICATION entre le serveur principal et le serveur de réPLICATION.
Scénario Alternative	Si une étape est saute ou mal configurer, la réPLICATION échouera
Post condition	<ul style="list-style-type: none"> • La réPLICATION entre le serveur principal et le serveur de réPLICATION est configurée et opérationnelle. • Les données de la base de données PostgreSQL sont répliquées en temps réel sur le serveur de réPLICATION.

Tableau 1 : Configuration réPLICATION

Titre	Gérer Utilisateur
Description/objectif	Ce cas d'utilisation représente la gestion d'utilisateurs dans un système
Acteur	Administrateur
Précondition	L'acteur est authentifié dans le système avec des privilèges administratifs ou de gestion des utilisateurs.
Scénario nominale	<ol style="list-style-type: none"> 1. L'acteur accède à l'interface de gestion des utilisateurs. 2. L'acteur sélectionne l'une des opérations suivantes : <ul style="list-style-type: none"> • [Créer un Utilisateur] • [Modifier un Utilisateur] • [Supprimer un Utilisateur] 3. Le système exécute le cas d'utilisation correspondant en fonction du choix de l'acteur.
Scénario Alternative	<ol style="list-style-type: none"> 1. Après avoir commencé à remplir le formulaire de création, l'acteur décide d'annuler la création de l'utilisateur. 2. L'acteur clique sur le bouton "Annuler" ou une option similaire. 3. Le système revient à l'interface principale de gestion des utilisateurs sans créer l'utilisateur.
Post condition	<ul style="list-style-type: none"> • Si l'utilisateur a soumis avec succès le formulaire de création, un nouvel utilisateur est créé dans le système avec les informations fournies. • Si l'utilisateur a annulé la création, aucun nouvel utilisateur n'est ajouté, et l'état du système reste inchangé.

Tableau 2 :Gérer Utilisateur

Titre	Effectuer une Transaction
Description/objectif	Ce cas d'utilisation représente l'exécution d'une transaction sur une base de données PostgreSQL.
Acteur	Utilisateur ou Administrateur
Précondition	L'utilisateur est connecté à la base de données PostgreSQL
Scénario nominale	<ol style="list-style-type: none"> 1. L'utilisateur démarre une transaction en choisissant "Commencer une Transaction" depuis l'interface de la base de données. 2. Le système démarre la transaction 3. L'utilisateur peut maintenant exécuter des requêtes SQL spécifiques à cette transaction. 4. Après avoir exécuté toutes les requêtes nécessaires, l'utilisateur choisit "Valider la Transaction." 5. Le système vérifie la validité des requêtes et, si tout est correct, confirme la transaction. 6. Les modifications apportées par la transaction sont enregistrées de manière permanente dans la base de données. 7. L'utilisateur reçoit une confirmation de la validation de la transaction.
Scénario Alternative	<ul style="list-style-type: none"> • Le système détecte une erreur ou une violation de contrainte lors de la validation • À tout moment avant la validation, l'utilisateur choisit d'annuler la transaction en sélectionnant "Annuler la Transaction" depuis l'interface
Post condition	Si l'utilisateur a choisi de valider la transaction avec succès, toutes les modifications apportées par les requêtes SQL de la transaction sont enregistrées de manière permanente dans la base de données.

Tableau 3 : Effectuer une Transaction

Titre	Surveiller la réPLICATION
Description/objectif	Ce cas d'utilisation permet à un administrateur ou à un utilisateur autorisé de surveiller l'état et la performance de la réPLICATION dans un environnement PostgreSQL.
Acteur	Utilisateur ou Administrateur
Précondition	<ul style="list-style-type: none"> • L'utilisateur est authentifié dans le système de surveillance de la réPLICATION PostgreSQL. • La réPLICATION PostgreSQL est configurée avec au moins un serveur principal et un serveur d'esclave.
Scénario nominale	<ol style="list-style-type: none"> 1. L'utilisateur accède à l'interface de surveillance de la réPLICATION PostgreSQL. 2. Le système affiche les informations de statut de la réPLICATION actuelle, y compris les serveurs impliqués, l'état de la réPLICATION (actif, inactif), la latence de réPLICATION, etc. 3. L'utilisateur peut sélectionner un serveur particulier pour obtenir des détails spécifiques sur son état de réPLICATION. 4. Le système peut afficher des graphiques ou des graphiques pour visualiser la latence de réPLICATION au fil du temps. 5. L'utilisateur peut configurer des alertes pour être informé en cas de problèmes de réPLICATION, tels que des retards excessifs. 6. L'utilisateur peut également effectuer des actions de maintenance, telles que la resynchronisation manuelle des serveurs d'esclaves en cas de besoin.
Scénario Alternative	Si des problèmes de réPLICATION sont détectés (par exemple, des retards excessifs ou une réPLICATION interrompue), l'utilisateur peut choisir de prendre des mesures correctives, telles que la réinitialisation de la réPLICATION.
Post condition	L'utilisateur est informé de l'état actuel de la réPLICATION PostgreSQL.

Tableau 4 : Surveiller la réPLICATION

Figure 7 : diagramme de classe

2.2.4 Conclusion

Dans ce chapitre, nous avons entrepris une analyse approfondie des besoins de l'entreprise SENELEC en matière de gestion de base de données, ainsi que la modélisation de la base de données PostgreSQL pour répondre à ces besoins.

3 **CHAPITRE III : IMPLEMENTATION ET PRESENTATION DES RESULTATS**

Dans ce chapitre consacré à l'implémentation et à la présentation de PostgreSQL, nous présentons la solution que nous avons adoptée, nous parlons d'abord des choix des outils et technologies utilisés dans notre projet ainsi que la présentation de l'application.

3.1 Les outils et technologies utilisés

- Ubuntu : C'est un système d'exploitation basé sur Linux. Il est open-source et largement utilisé, offrant une interface conviviale pour les utilisateurs. Ubuntu est connu pour sa stabilité, sa sécurité et sa grande compatibilité avec une variété de matériels et de logiciels. Il est populaire tant pour une utilisation personnelle que pour des serveurs et des environnements de développement.

Nous allons installer PostgreSQL sur Ubuntu (V 22.04.2).^[4]



Figure 8 : logo Ubuntu

- Virtual-Box : VirtualBox est un puissant produit de virtualisation x86 et AMD64/Intel64 destiné à une utilisation en entreprise et à domicile. VirtualBox est non seulement un produit extrêmement riche en fonctionnalités et hautes performances destiné aux entreprises clientes, mais il s'agit également de la seule solution professionnelle disponible gratuitement en tant que logiciel Open Source selon les termes de la licence publique générale GNU (GPL) version 3.^[5]



Figure 9 : logo Virtual Box

3.2 Présentation de PostgreSQL

PostgreSQL est un système de gestion de base de données relationnelle orientée objet puissant et open source qui est capable de prendre en charge en toute sécurité les charges de travail de données les plus complexes. Alors que MySQL donne la priorité à l'évolution et aux performances, PostgreSQL donne la priorité à la conformité et à l'extensibilité SQL.^[6]



Figure 10 : logo PostgreSQL

3.2.1 Pourquoi utiliser PostgreSQL ?

3.2.1.1 Avantages

Voici une liste des avantages de PostgreSQL :

- Fiabilité et stabilité : Conçu pour gérer de grandes quantités de données et fonctionne sans interruption pendant de longues périodes.
- Conformité aux normes SQL et ACID : Prend en charge une large gamme de fonctionnalités et de syntaxes SQL. Il suit les normes ACID (Atomicité, Cohérence, Isolation, Durabilité), garantissant l'intégrité et la cohérence des données.
- Bonne performance : En particulier pour les requêtes complexes et les charges de travail lourdes. Il dispose d'un puissant optimiseur de requêtes qui analyse les requêtes et génère des plans d'exécution efficaces pour accélérer le traitement de données.
- Communauté open source : Dynamique et engagée, la communauté fournit un soutien, des mises à jour régulières, des correctifs de sécurité et un écosystème riche de modules complémentaires et d'outils développés par des tiers.
- Sécurité : Offre des fonctionnalités de sécurité robustes, notamment l'authentification, les autorisations granulaires, le chiffrement des données, la journalisation des accès et la gestion des certificats SSL/TTL.
- Compatibilité : PostgreSQL est conçu pour être compatible avec une large gamme de systèmes

d'exploitation (SE).^[7]

3.2.1.2 Inconvénients

- Complexité : Elle est plus complexe à configurer et à gérer par rapport à des systèmes de gestion de bases de données plus simples. Il peut nécessiter une expertise technique pour optimiser les performances, gérer les réplications et effectuer les opérations de maintenance.
- Consommation en ressources système que certaines autres bases de données lorsqu'il est configuré pour gérer de gros volumes de données. Cela peut nécessiter des ressources matérielles plus importantes.
- Pic de performance : Bien qu'elle offre de bonnes performances dans la plupart des cas, il peut être moins performant dans certaines situations, notamment avec des charges de travail spécifiques qui nécessitent des opérations de lecture intensive ou de traitement de données massives.
- Comparé à certains autres SGBD, PostgreSQL peut avoir un écosystème d'outils et de plugins moins étendu. Cela peut rendre la recherche d'outils spécifiques ou de fonctionnalités tierces plus limitées.^[8]

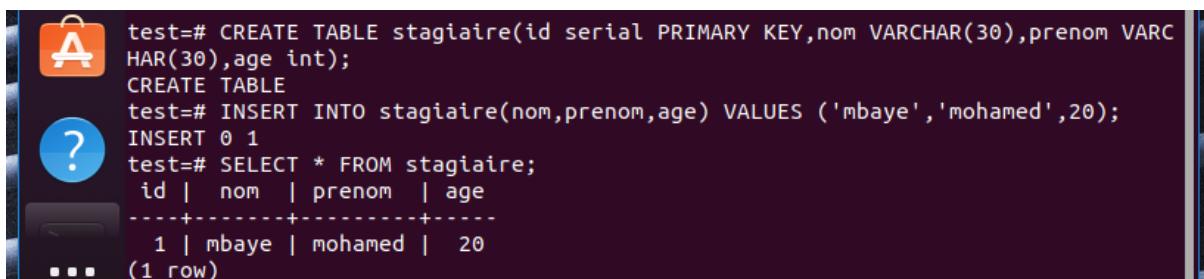
3.2.2 Schéma de la base de données

Le schéma suivant représente la base de données 'test' ainsi que la 'stagiaire' a répliqué.



```
postgres=# create database test;
CREATE DATABASE
postgres=#
```

Figure 11 : base de donnée 'test'



```
test=# CREATE TABLE stagiaire(id serial PRIMARY KEY,nom VARCHAR(30),prenom VARCHAR(30),age int);
CREATE TABLE
test=# INSERT INTO stagiaire(nom,prenom,age) VALUES ('mbaye','mohamed',20);
INSERT 0 1
test=# SELECT * FROM stagiaire;
 id | nom   | prenom | age
----+-----+-----+
  1 | mbaye | mohamed | 20
(1 row)
```

Figure 12 : la table 'stagiaire'

3.2.3 Installation

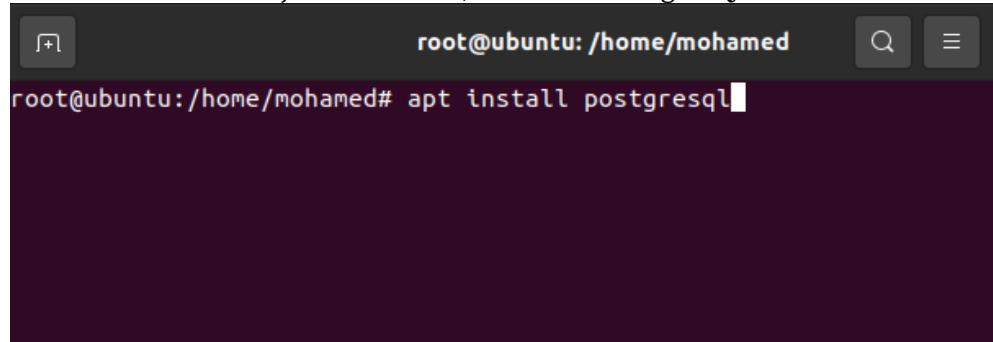
Ouvrir un terminal avec la combinaison **Ctrl + Alt + T**

Mettez à jour les paquets disponibles en exécutant la commande suivante :

```
mohamed@ubuntu:~$ sudo -s
[sudo] password for mohamed:
root@ubuntu:/home/mohamed# apt update
```

Figure 13 : Mettre à jour les paquets

Une fois la mise à jour terminée, installez PostgreSQL en utilisant la commande suivante :

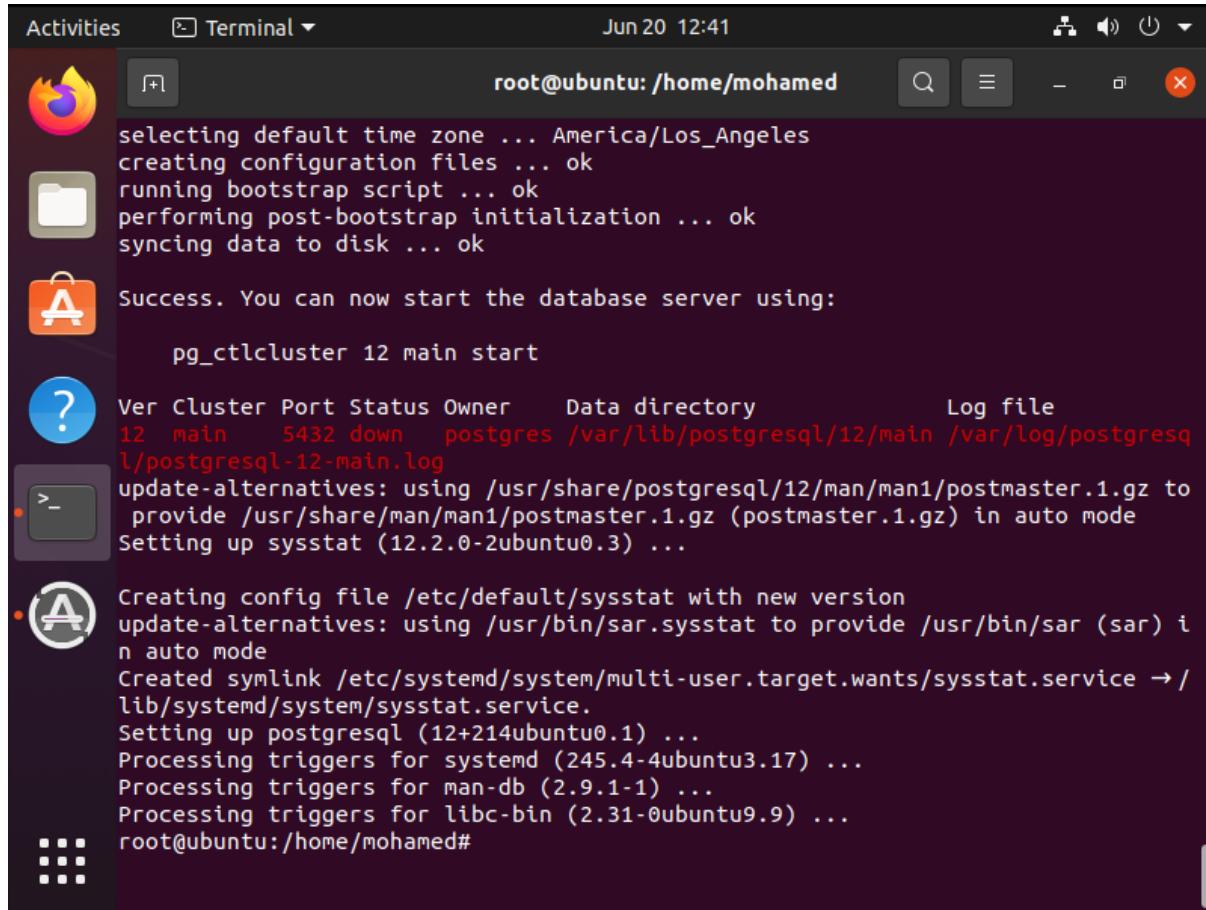


```
Reading state information... Done
The following additional packages will be installed:
  libllvm10 libpq5 postgresql-12 postgresql-client-12
  postgresql-client-common postgresql-common sysstat
Suggested packages:
  postgresql-doc postgresql-doc-12 libjson-perl isag
The following NEW packages will be installed:
  libllvm10 libpq5 postgresql postgresql-12 postgresql-client-12
  postgresql-client-common postgresql-common sysstat
0 upgraded, 8 newly installed, 0 to remove and 254 not upgraded.
Need to get 15.1 MB/30.6 MB of archives.
After this operation, 121 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libpq5 amd64
  12.15-0ubuntu0.20.04.1 [116 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 postgresql-c
lient-12 amd64 12.15-0ubuntu0.20.04.1 [1,052 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 postgresql-1
2 amd64 12.15-0ubuntu0.20.04.1 [13.5 MB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 sysstat amd6
4 12.2.0-2ubuntu0.3 [448 kB]
Fetched 15.1 MB in 6s (2,583 kB/s)
Preconfiguring packages ...
Selecting previously unselected package libllvm10:amd64.
(Reading database ... 121327 files and directories currently installed.)
Preparing to unpack .../0-libllvm10_1%3a10.0.0-4ubuntu1_amd64.deb ...
Unpacking libllvm10:amd64 (1:10.0.0-4ubuntu1) ...

Progress: [  3%] [#.....]
```

Figure 14 : installer PostgreSQL

Nous avons en rouge les informations de l'installation de PostgreSQL tel que le Cluster, le Port, le répertoire de données etc...



```

Activities Terminal ▾ Jun 20 12:41
root@ubuntu: /home/mohamed
[+]
selecting default time zone ... America/Los_Angeles
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok

Success. You can now start the database server using:

    pg_ctlcluster 12 main start

Ver Cluster Port Status Owner      Data directory          Log file
12  main      5432 down    postgres /var/lib/postgresql/12/main /var/log/postgresql/postgresql-12-main.log
update-alternatives: using /usr/share/postgresql/12/man/man1/postmaster.1.gz to
                     provide /usr/share/man/man1/postmaster.1.gz (postmaster.1.gz) in auto mode
Setting up sysstat (12.2.0-2ubuntu0.3) ...

Creating config file /etc/default/sysstat with new version
update-alternatives: using /usr/bin/sar.sysstat to provide /usr/bin/sar (sar) in auto mode
Created symlink /etc/systemd/system/multi-user.target.wants/sysstat.service → /lib/systemd/system/sysstat.service.
Setting up postgresql (12+214ubuntu0.1) ...
Processing triggers for systemd (245.4-4ubuntu3.17) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
root@ubuntu:/home/mohamed#

```

Figure 15 : informations de l'installation de PostgreSQL

- **VER** : Affiche la version de PostgreSQL.
- **CLUSTER** : Fait référence à un ensemble de bases de données gérées par une seule instance de serveur PostgreSQL.
- **PORT** : Ce paramètre fait référence au numéro de port utilisé par le cluster PostgreSQL.
- **STATUS** : Cette partie de la commande nous permet d'obtenir l'état actuel du cluster PostgreSQL.
- **OWNER** : Il s'agit de l'utilisateur propriétaire du cluster PostgreSQL. C'est l'utilisateur qui a les droits de contrôle sur le cluster et qui peut effectuer des opérations d'administration.
- **DATA DIRECTORY** : Cette partie de la commande indique le répertoire où sont stockées les données du cluster PostgreSQL.

- **LOG FILE** : Il s'agit du fichier de journal (log) du cluster PostgreSQL.

Une fois l'installation terminée, PostgreSQL sera automatiquement démarré et configuré pour se lancer au démarrage du système.

3.2.4 Configuration

Nous devons définir un mot de passe pour le superutilisateur en suivant les étapes suivantes :

-Se connecter en tant que superutilisateur dans PostgreSQL et définir un mot de passe pour le superutilisateur.

```
root@ubuntu:/home/mohamed# sudo -u postgres psql
psql (12.15 (Ubuntu 12.15-0ubuntu0.20.04.1))
Type "help" for help.

postgres=# ALTER USER postgres WITH PASSWORD 'Senelec2003#';
ALTER ROLE
postgres=# \q
root@ubuntu:/home/mohamed#
```

Figure 16 :Ajout du mot de passe superutilisateur

-Ouvrons le fichier de configuration de PostgreSQL en utilisant un éditeur de texte ,recherchons les lignes qui contiennent le motif suivant **local all** et modifiez la méthode d'authentification de '**peer**' à '**md5**'.

```
root@ubuntu:/home/mohamed# nano /etc/postgresql/12/main/pg_hba.conf █
```

```
# Database administrative login by Unix domain socket
local    all            postgres                                peer
# TYPE   DATABASE        USER            ADDRESS             METHOD
# "local" is for Unix domain socket connections only
local    all            all                                     peer
# IPv4 local connections:
host     all            all            127.0.0.1/32      md5
# IPv6 local connections:
host     all            all            ::1/128           md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication   all                                     peer
host   replication   all            127.0.0.1/32      md5
host   replication   all            ::1/128           md5
```

```
# Database administrative login by Unix domain socket
local    all            postgres                                md5
# TYPE   DATABASE        USER            ADDRESS             METHOD
# "local" is for Unix domain socket connections only
local    all            all                                     md5
# IPv4 local connections:
host     all            all            127.0.0.1/32      md5
# IPv6 local connections:
host     all            all            ::1/128           md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local   replication   all                                     md5
host   replication   all            127.0.0.1/32      md5
host   replication   all            ::1/128           md5
```

Figure 17 : Modification méthode de connexion

Cela indique à PostgreSQL d'utiliser une authentification basée sur un mot de passe pour le superutilisateur "postgres".

Redémarrer le service PostgreSQL pour appliquer les modifications en utilisant la commande suivante :

```
root@ubuntu:/home/mohamed# systemctl restart postgresql
```

Figure 18 :Redémarre service PostgreSQL

Maintenant lorsqu'on se connecte un mot de passe est requis.

```
root@ubuntu:/home/mohamed# sudo -u postgres psql  
Password for user postgres: [REDACTED]
```

Figure 19 : Connexion avec mot de passe

Nous passons à l'installation graphique de PostgreSQL **pgadmin4**.

Allons sur le site de téléchargement : [Community DL Page \(enterprisedb.com\)](https://www.postgresql.org/download)

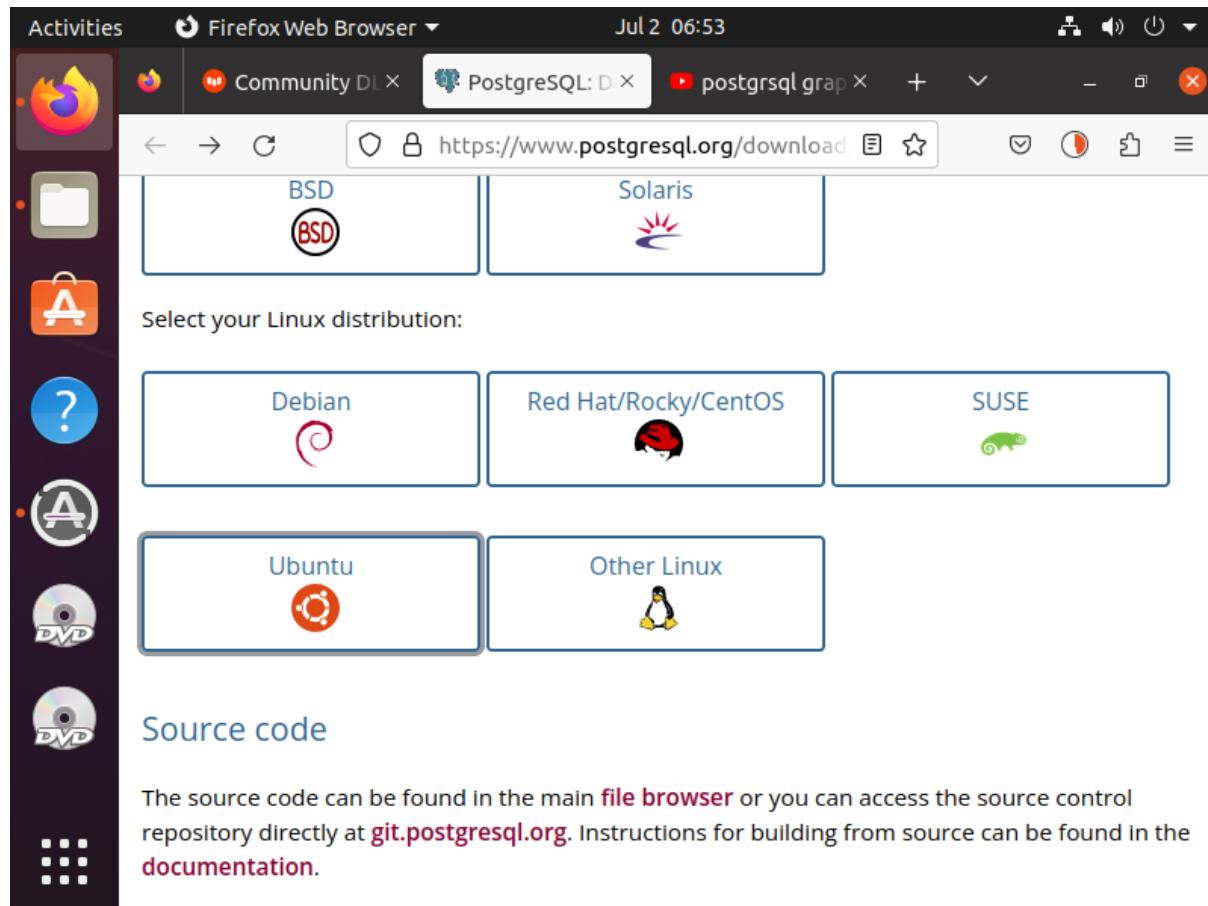


Figure 20 : Site de téléchargement de PostgreSQL

Choisissons **Ubuntu** et copions le script.

Un script shell est un fichier contenant des commandes shell, écrit dans un langage de script shell comme Bash. Il est utilisé pour automatiser des tâches, exécuter des commandes système ou lancer des programmes

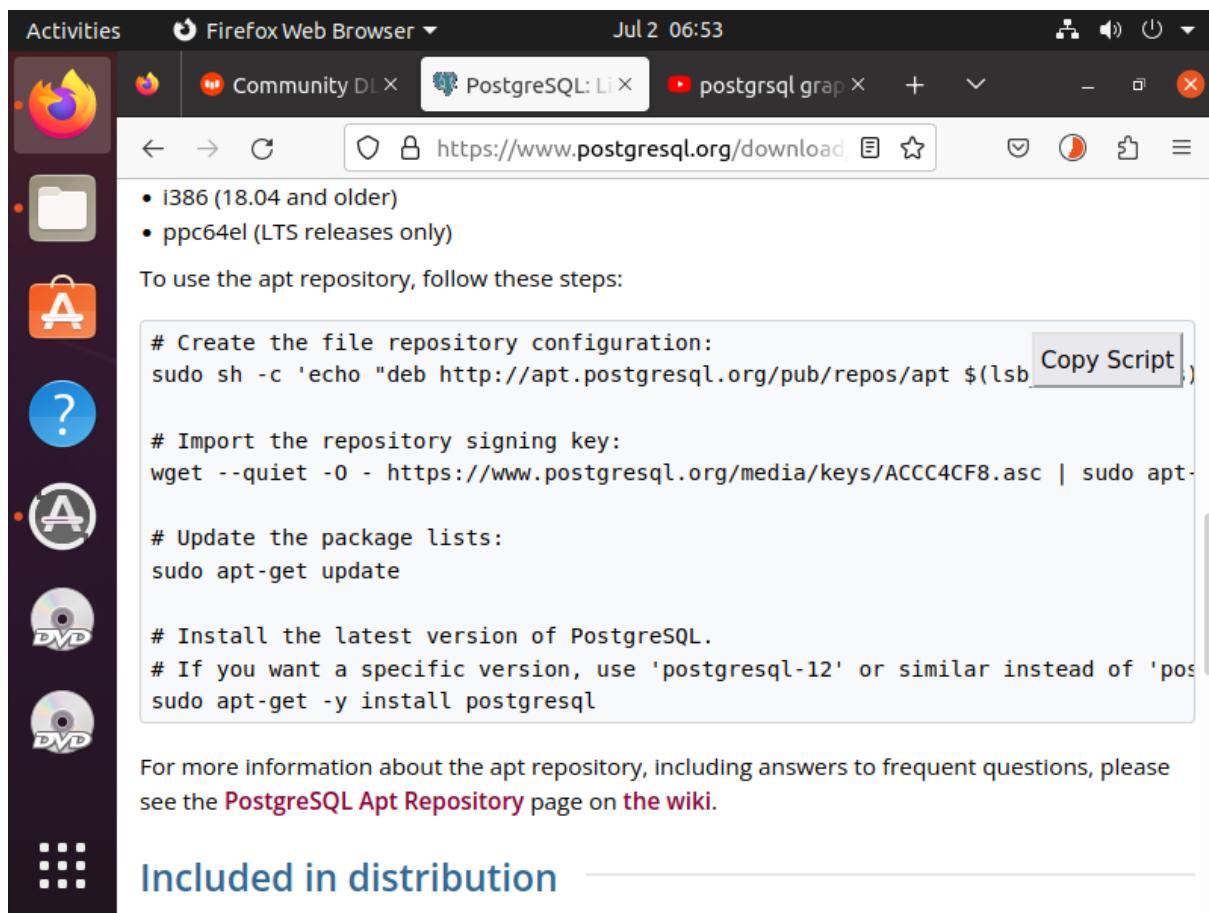
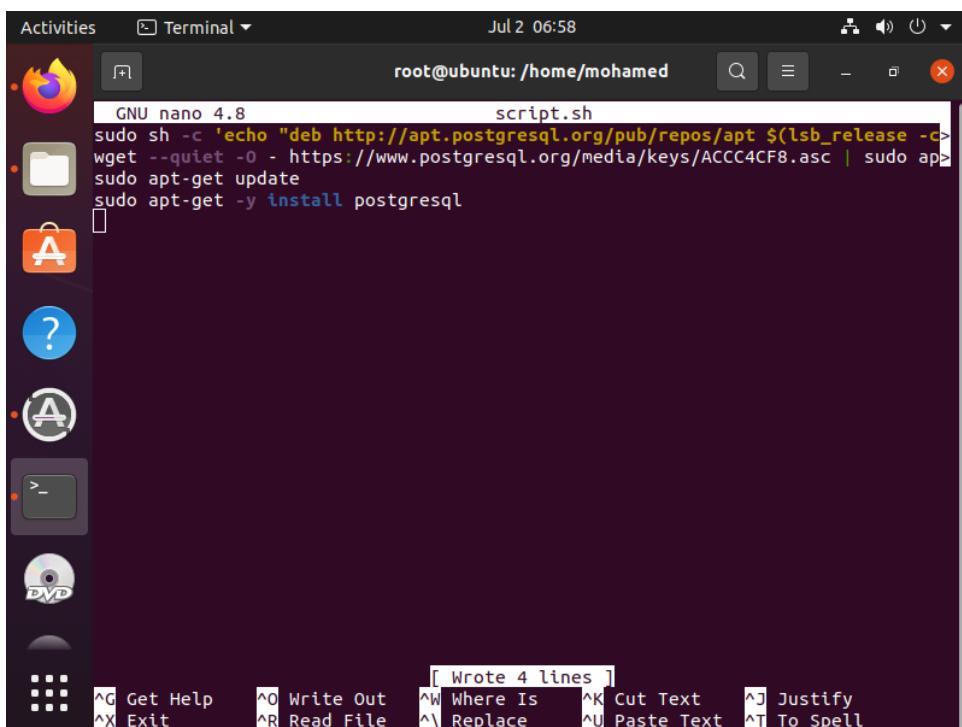
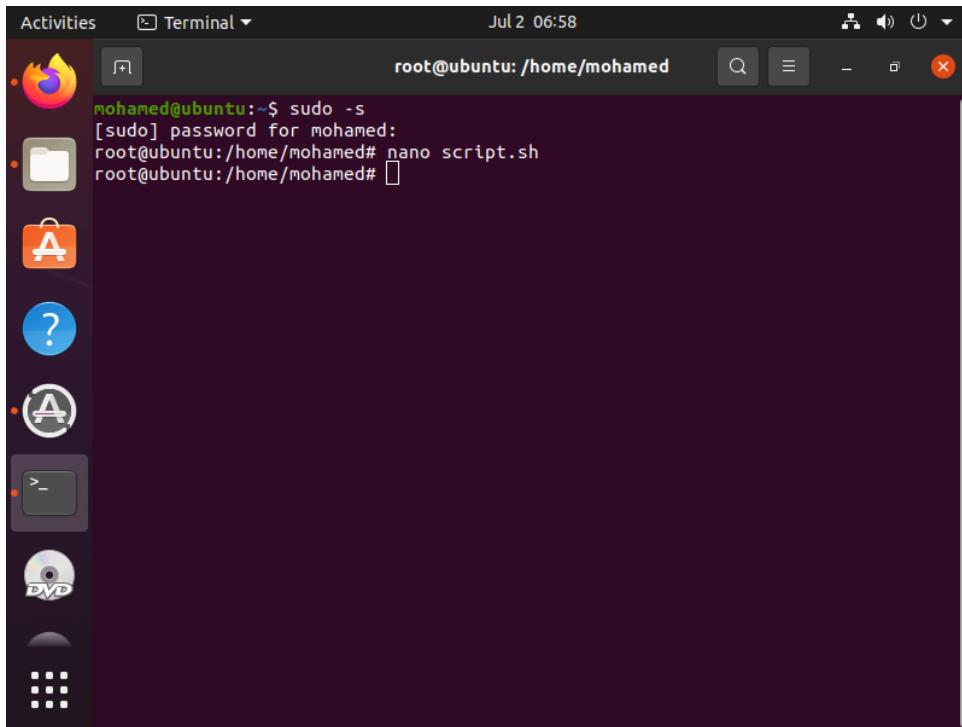


Figure 21 : Script à copier

Ouvrons un terminal, éditons un fichier .sh et collons le script dans le fichier .sh



Fermons et exécutons le script.

The image shows two terminal windows side-by-side, both titled "Terminal".

Top Terminal (Jul 2 07:00):

```
mohamed@ubuntu:~$ sudo -s
[sudo] password for mohamed:
root@ubuntu:/home/mohamed# nano script.sh
root@ubuntu:/home/mohamed# nano script.sh
root@ubuntu:/home/mohamed# . script.sh
OK
Get:1 http://security.ubuntu.com/ubuntu focal-security InRelease [114 kB]
Get:2 http://apt.postgresql.org/pub/repos/apt focal-pgdg InRelease [123 kB]
Hit:3 http://us.archive.ubuntu.com/ubuntu focal InRelease
Get:4 http://us.archive.ubuntu.com/ubuntu focal-updates InRelease [114 kB]
Get:5 http://apt.postgresql.org/pub/repos/apt focal-pgdg/main amd64 Packages [273 kB]
Get:6 http://us.archive.ubuntu.com/ubuntu focal-backports InRelease [108 kB]
Get:7 http://us.archive.ubuntu.com/ubuntu focal-updates/restricted amd64 Packages [2,061 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu focal-updates/restricted i386 Packages [32.1 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse amd64 Packages [31.1 kB]
Get:10 http://us.archive.ubuntu.com/ubuntu focal-updates/multiverse i386 Packages [9,888 B]
Fetched 2,867 kB in 4s (784 kB/s)
Reading package lists... 5%
```

Bottom Terminal (Jul 2 07:03):

```
n/pgdg/apt.postgresql.org.gpg'
Setting up postgresql-15 (15.3-1.pgdg20.04+1) ...
Creating new PostgreSQL cluster 15/main ...
/usr/lib/postgresql/15/bin/initdb -D /var/lib/postgresql/15/main --auth-local peer --auth-host scram-sha-256 --no-instructions
The files belonging to this database system will be owned by user "postgres".
This user must also own the server process.

The database cluster will be initialized with locale "en_US.UTF-8".
The default database encoding has accordingly been set to "UTF8".
The default text search configuration will be set to "english".

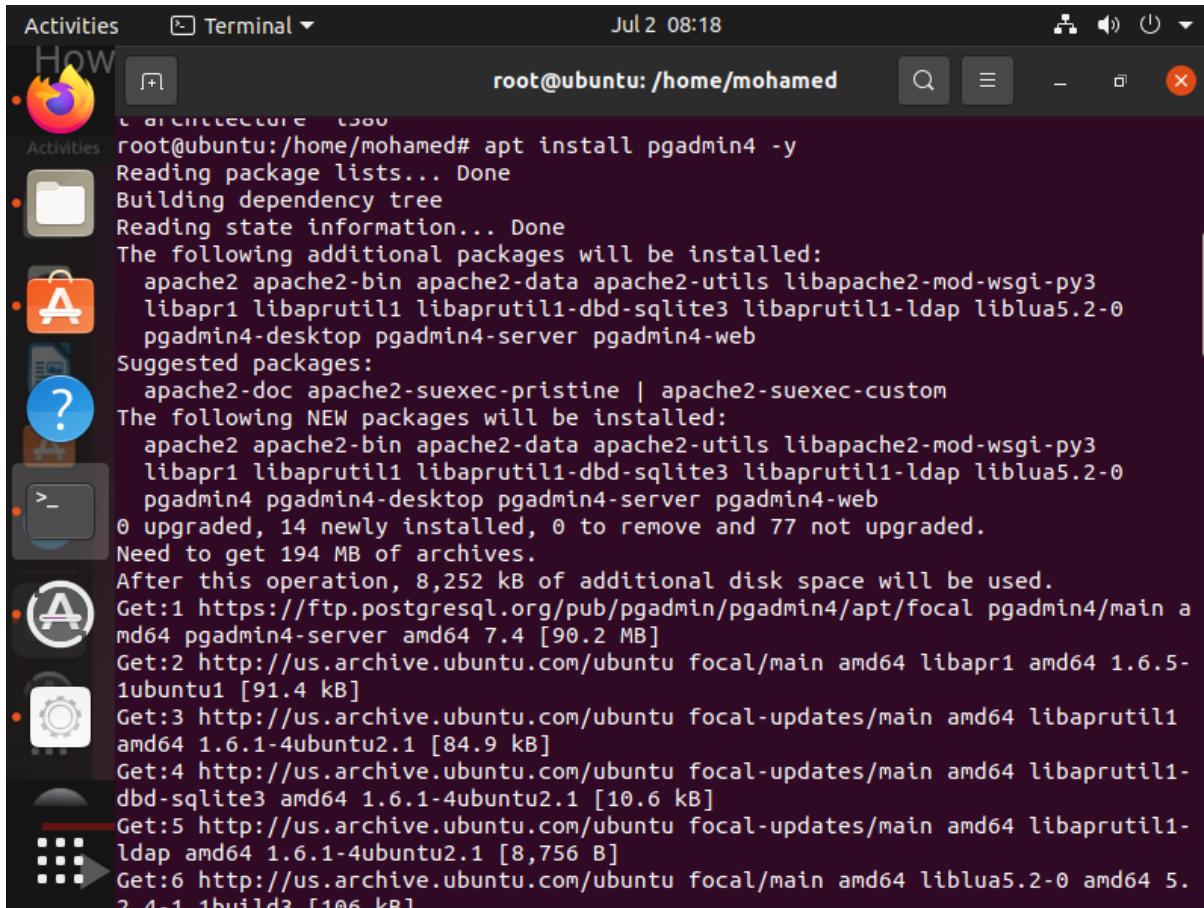
Data page checksums are disabled.

fixing permissions on existing directory /var/lib/postgresql/15/main ... ok
creating subdirectories ... ok
selecting dynamic shared memory implementation ... posix
selecting default max_connections ... 100
selecting default shared_buffers ... 128MB
selecting default time zone ... America/Los_Angeles
creating configuration files ... ok
running bootstrap script ... ok
performing post-bootstrap initialization ... ok
syncing data to disk ... ok
Setting up postgresql (15+250.pgdg20.04+1) ...
Processing triggers for systemd (245.4-4ubuntu3.20) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.9) ...
root@ubuntu:/home/mohamed# 
```

Figure 22: exécution du script

Installons l'administrateur graphique de PostgreSQL via la commande :

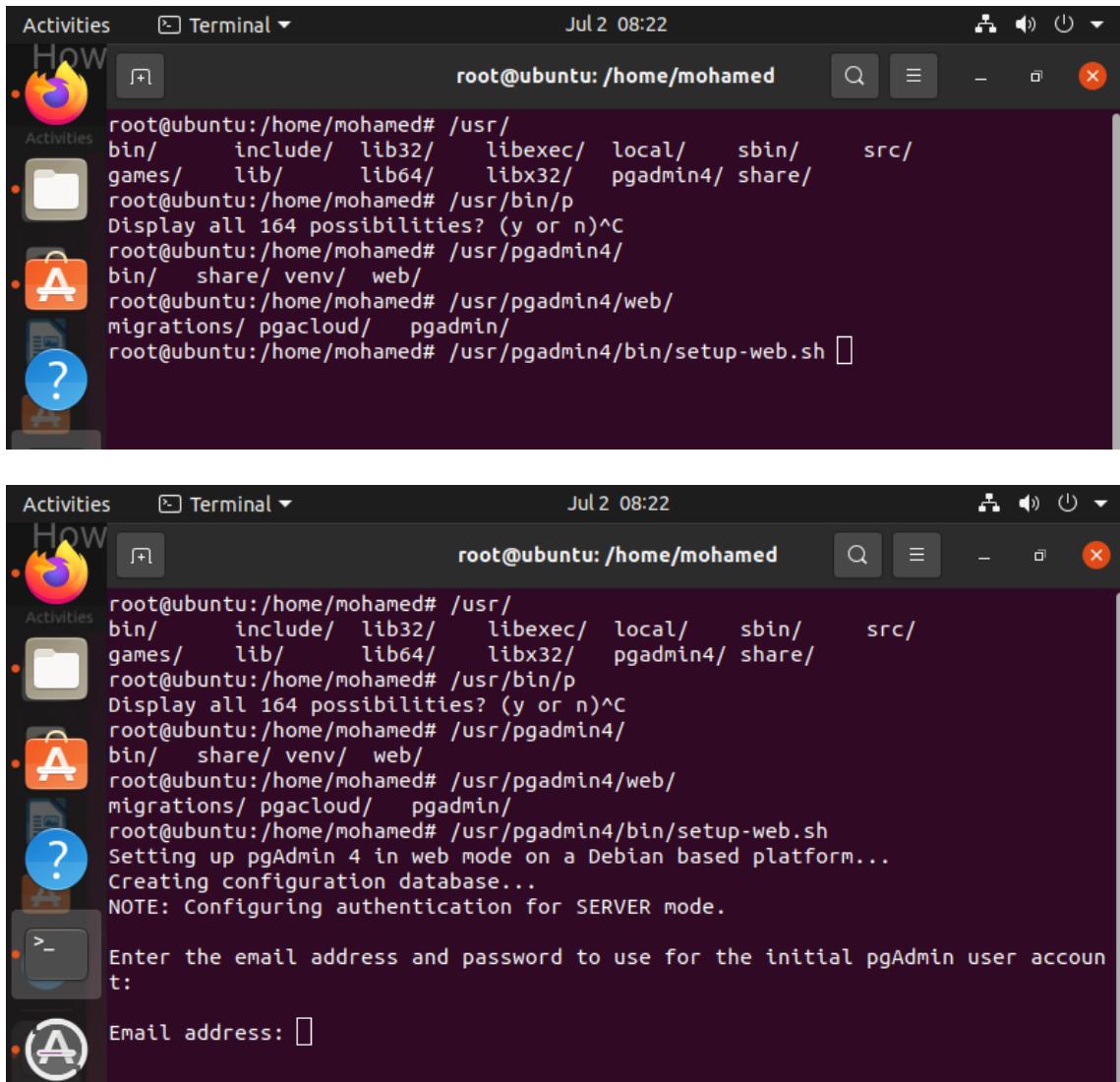
```
apt install pgadmin4 -y
```



```
root@ubuntu:/home/mohamed# apt install pgadmin4 -y
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapache2-mod-wsgi-py3
  libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
  pgadmin4-desktop pgadmin4-server pgadmin4-web
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils libapache2-mod-wsgi-py3
  libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.2-0
  pgadmin4 pgadmin4-desktop pgadmin4-server pgadmin4-web
0 upgraded, 14 newly installed, 0 to remove and 77 not upgraded.
Need to get 194 MB of archives.
After this operation, 8,252 kB of additional disk space will be used.
Get:1 https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/focal pgadmin4/main amd64 pgadmin4-server amd64 7.4 [90.2 MB]
Get:2 http://us.archive.ubuntu.com/ubuntu focal/main amd64 libapr1 amd64 1.6.5-1ubuntu1 [91.4 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libaprutil1 amd64 1.6.1-4ubuntu2.1 [84.9 kB]
Get:4 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-4ubuntu2.1 [10.6 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu focal-updates/main amd64 libaprutil1-ldap amd64 1.6.1-4ubuntu2.1 [8,756 B]
Get:6 http://us.archive.ubuntu.com/ubuntu focal/main amd64 liblua5.2-0 amd64 5.2.4-1~build3 [106 kB]
```

Figure 23 : installation de pgAdmin4

Après l'installation, allons dans le dossier '**/usr/pgadmin4/bin/**' et exécutons le script '**setup-web.sh**'.



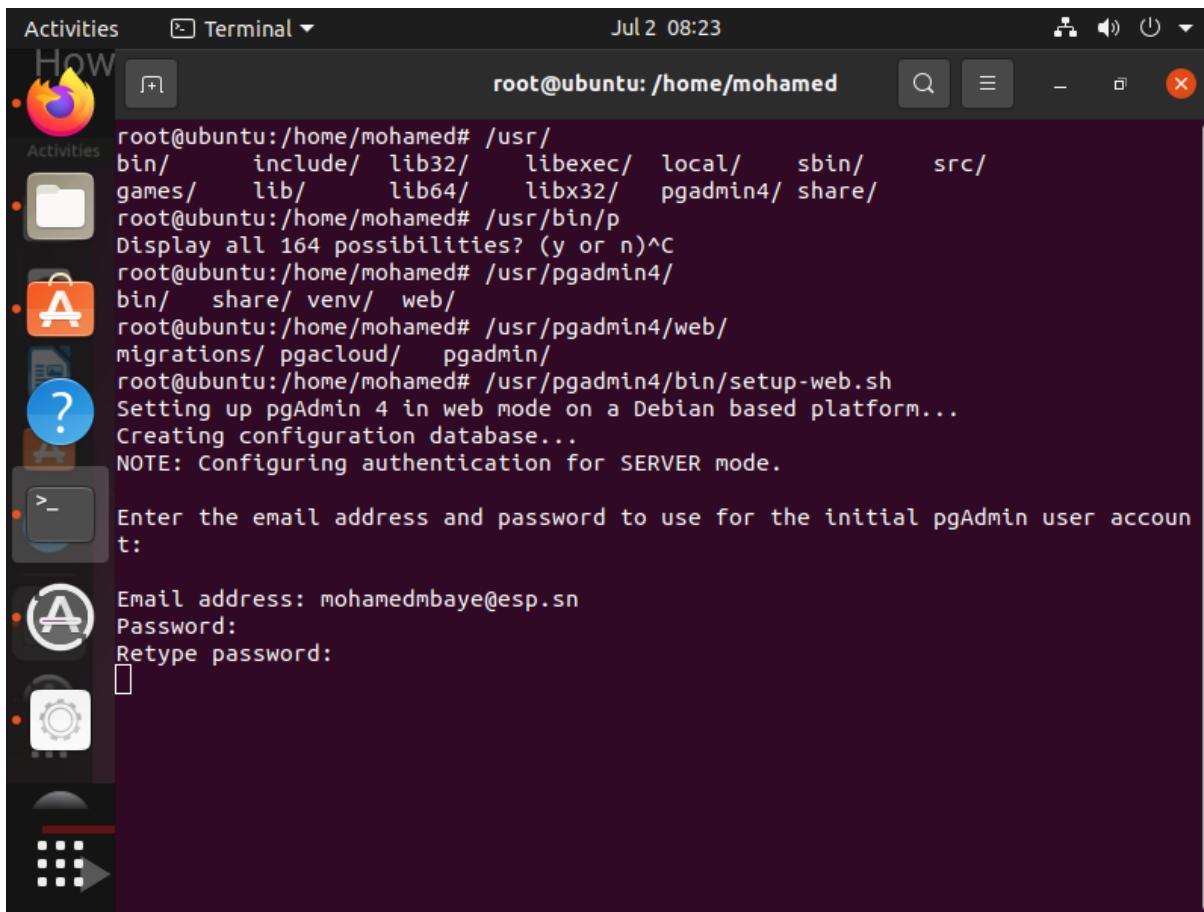
```
root@ubuntu:/home/mohamed# /usr/
bin/ include/ lib32/ libexec/ local/ sbin/ src/
games/ lib/ lib64/ libx32/ pgadmin4/ share/
root@ubuntu:/home/mohamed# /usr/bin/p
Display all 164 possibilities? (y or n)^C
root@ubuntu:/home/mohamed# /usr/pgadmin4/
bin/ share/ venv/ web/
root@ubuntu:/home/mohamed# /usr/pgadmin4/web/
migrations/ pgacloud/ pgadmin/
root@ubuntu:/home/mohamed# /usr/pgadmin4/bin/setup-web.sh
```

```
root@ubuntu:/home/mohamed# /usr/
bin/ include/ lib32/ libexec/ local/ sbin/ src/
games/ lib/ lib64/ libx32/ pgadmin4/ share/
root@ubuntu:/home/mohamed# /usr/bin/p
Display all 164 possibilities? (y or n)^C
root@ubuntu:/home/mohamed# /usr/pgadmin4/
bin/ share/ venv/ web/
root@ubuntu:/home/mohamed# /usr/pgadmin4/web/
migrations/ pgacloud/ pgadmin/
root@ubuntu:/home/mohamed# /usr/pgadmin4/bin/setup-web.sh
Setting up pgAdmin 4 in web mode on a Debian based platform...
Creating configuration database...
NOTE: Configuring authentication for SERVER mode.

Enter the email address and password to use for the initial pgAdmin user account:
Email address:
```

Figure 24 : exécution du script 'setup-web.sh'

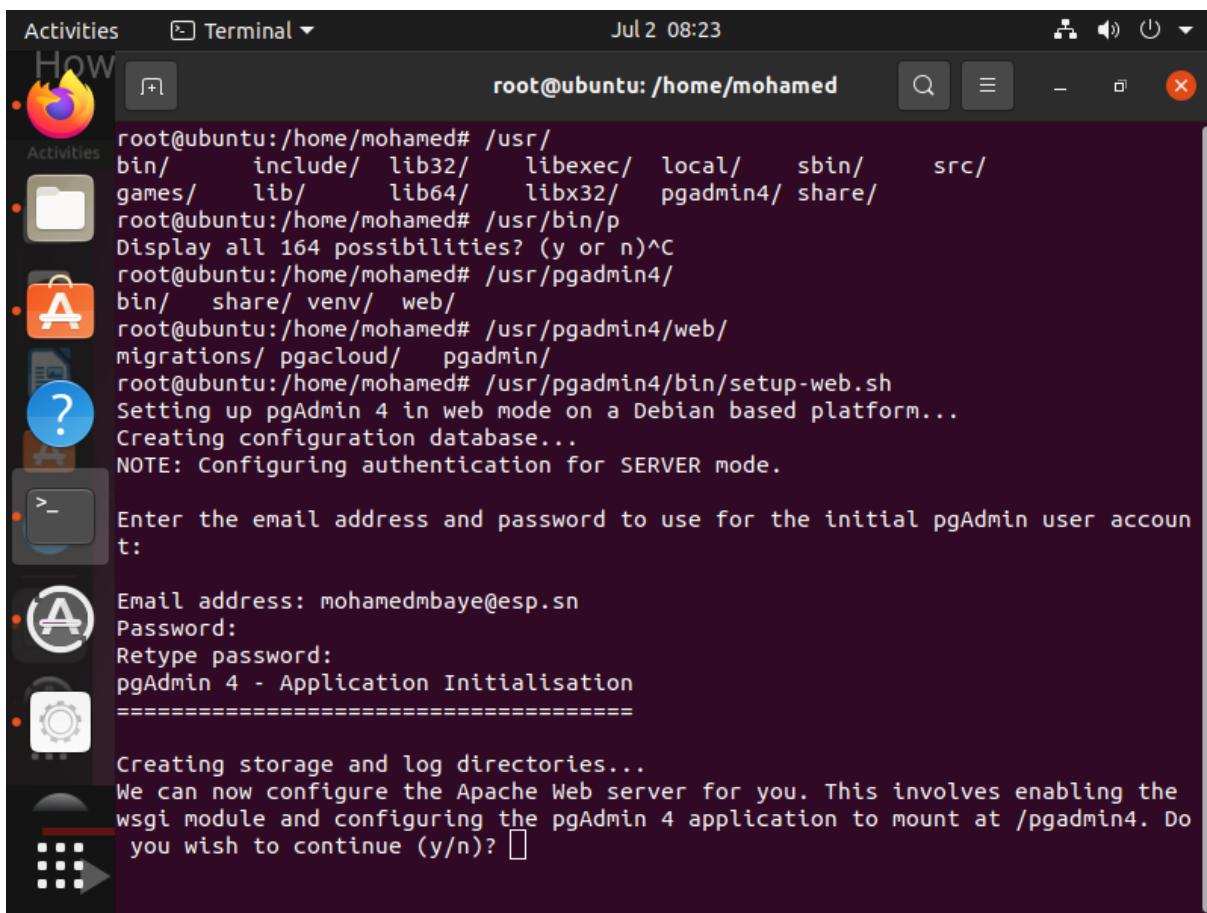
Nous devons entrer un e-mail pour le compte principal puis un mot de passe



```
root@ubuntu:/home/mohamed# /usr/
bin/      include/  lib32/   libexec/  local/    sbin/    src/
games/    lib/       lib64/   libx32/   pgadmin4/ share/
root@ubuntu:/home/mohamed# /usr/bin/p
Display all 164 possibilities? (y or n)^C
root@ubuntu:/home/mohamed# /usr/pgadmin4/
bin/    share/ venv/  web/
root@ubuntu:/home/mohamed# /usr/pgadmin4/web/
migrations/ pgacloud/ pgadmin/
root@ubuntu:/home/mohamed# /usr/pgadmin4/bin/setup-web.sh
Setting up pgAdmin 4 in web mode on a Debian based platform...
Creating configuration database...
NOTE: Configuring authentication for SERVER mode.

Enter the email address and password to use for the initial pgAdmin user account:
Email address: mohamedmbaye@esp.sn
Password:
Retype password:
```

Figure 25 : Saisie du e-mail et mot de passe



The screenshot shows a terminal window titled "root@ubuntu: /home/mohamed" running on a Ubuntu desktop environment. The terminal displays the following command and its output:

```
root@ubuntu:/home/mohamed# /usr/
bin/      include/  lib32/  libexec/  local/  sbin/  src/
games/   lib/    lib64/  libx32/  pgadmin4/ share/
root@ubuntu:/home/mohamed# /usr/bin/p
Display all 164 possibilities? (y or n)^C
root@ubuntu:/home/mohamed# /usr/pgadmin4/
bin/    share/ venv/ web/
root@ubuntu:/home/mohamed# /usr/pgadmin4/web/
migrations/ pgacloud/ pgadmin/
root@ubuntu:/home/mohamed# /usr/pgadmin4/bin/setup-web.sh
Setting up pgAdmin 4 in web mode on a Debian based platform...
Creating configuration database...
NOTE: Configuring authentication for SERVER mode.

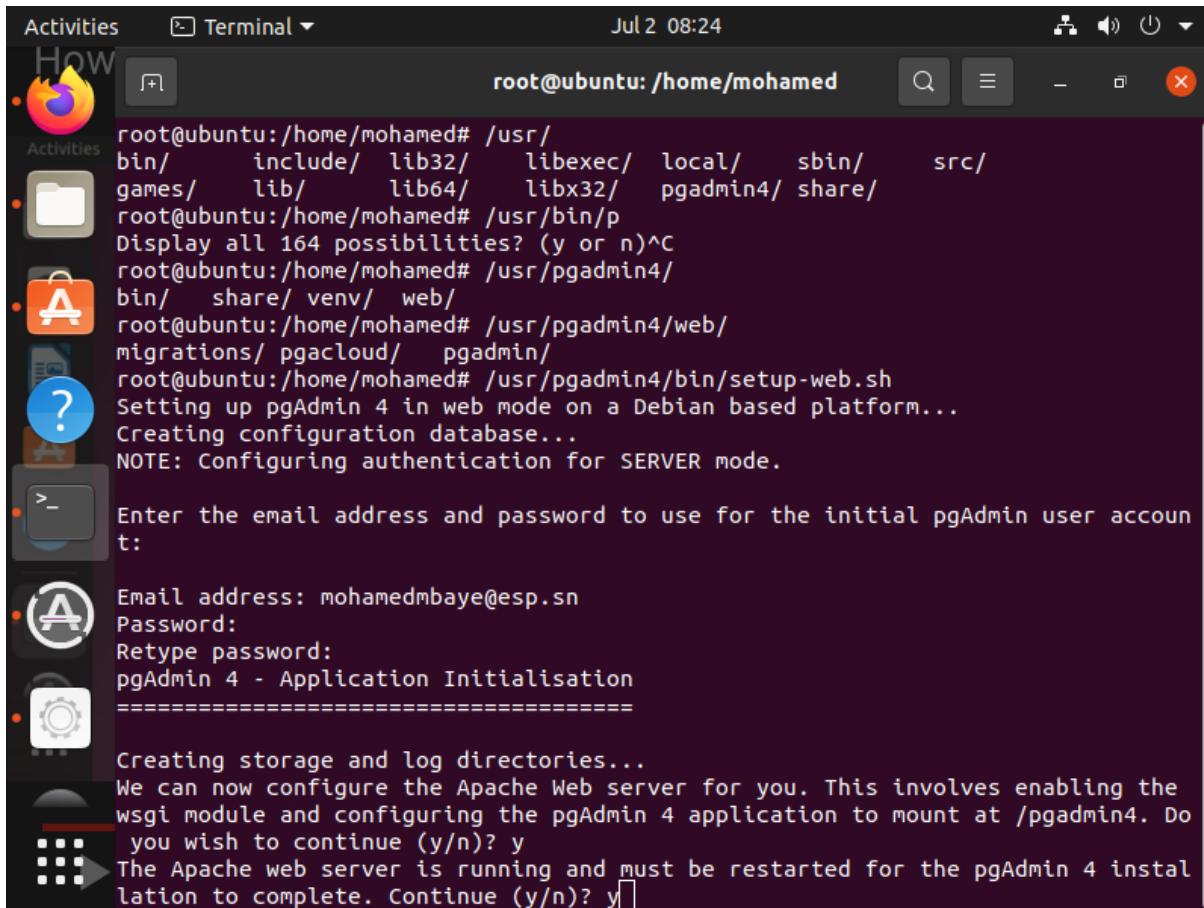
Enter the email address and password to use for the initial pgAdmin user account:

Email address: mohamedmbayed@esp.sn
Password:
Retype password:
pgAdmin 4 - Application Initialisation
=====
Creating storage and log directories...
We can now configure the Apache Web server for you. This involves enabling the
wsgi module and configuring the pgAdmin 4 application to mount at /pgadmin4. Do
you wish to continue (y/n)? [
```

Figure 26 :Initialisation de pgadmin4

Maintenant, pgAdmin 4 est initialisée avec la création des stockages et des fichiers d'erreurs.

Nous pouvons configurer Apache Web Server pour nous mais activerait le wsgi module et configurera pgAdmin 4 à **/pgadmin4**. Pour cela tapons **y** sinon **n** pour annuler.



```
root@ubuntu:/home/mohamed# /usr/
bin/      include/  lib32/    libexec/   local/    sbin/    src/
games/    lib/       lib64/    libx32/   pgadmin4/ share/
root@ubuntu:/home/mohamed# /usr/bin/p
Display all 164 possibilities? (y or n)^C
root@ubuntu:/home/mohamed# /usr/pgadmin4/
bin/    share/ venv/  web/
root@ubuntu:/home/mohamed# /usr/pgadmin4/web/
migrations/ pgacloud/ pgadmin/
root@ubuntu:/home/mohamed# /usr/pgadmin4/bin/setup-web.sh
Setting up pgAdmin 4 in web mode on a Debian based platform...
Creating configuration database...
NOTE: Configuring authentication for SERVER mode.

Enter the email address and password to use for the initial pgAdmin user account:
Email address: mohamedmbaye@esp.sn
Password:
Retype password:
pgAdmin 4 - Application Initialisation
=====
Creating storage and log directories...
We can now configure the Apache Web server for you. This involves enabling the
wsgi module and configuring the pgAdmin 4 application to mount at /pgadmin4. Do
you wish to continue (y/n)? y
The Apache web server is running and must be restarted for the pgAdmin 4 instal
lation to complete. Continue (y/n)? y
```

Figure 27 : installation de pgadmin4

Pour achever l'installation de pgAdmin 4 il faut redémarrer le serveur d'Apache Web.

```
root@ubuntu:/home/mohamed# /usr/pgadmin4/bin/setup-web.sh
Display all 164 possibilities? (y or n)^C
root@ubuntu:/home/mohamed# /usr/pgadmin4/
bin/ share/ venv/ web/
root@ubuntu:/home/mohamed# /usr/pgadmin4/web/
migrations/ pgacloud/ pgadmin/
root@ubuntu:/home/mohamed# /usr/pgadmin4/bin/setup-web.sh
Setting up pgAdmin 4 in web mode on a Debian based platform...
Creating configuration database...
NOTE: Configuring authentication for SERVER mode.

Enter the email address and password to use for the initial pgAdmin user account:
Email address: mohamedmbaye@esp.sn
Password:
Retype password:
pgAdmin 4 - Application Initialisation
=====
Creating storage and log directories...
We can now configure the Apache Web server for you. This involves enabling the
wsgi module and configuring the pgAdmin 4 application to mount at /pgadmin4. Do
you wish to continue (y/n)? y
The Apache web server is running and must be restarted for the pgAdmin 4 instal
lation to complete. Continue (y/n)? y
Apache successfully restarted. You can now start using pgAdmin 4 in web mode at
http://127.0.0.1/pgadmin4
root@ubuntu:/home/mohamed#
```

Figure 28 : fin de l'installation de pgadmin4

Le lien accès au site de pgAdmin 4 est : <https://127.0.0.1/pgadmin4>

Voici l'interface de connexion de pgAdmin 4, un identifiant et un mot de passe est requis.

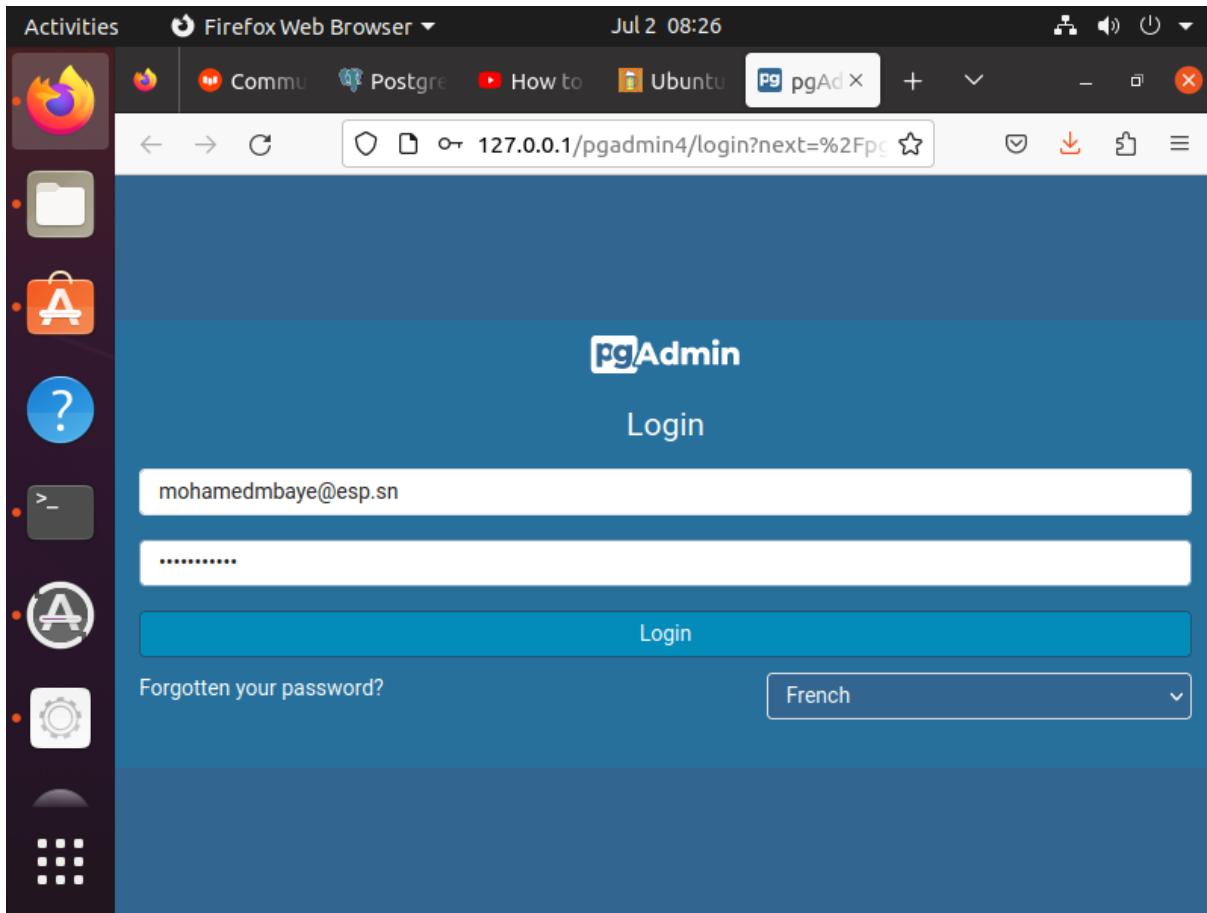


Figure 29: interface de connexion pgadmin4

Maintenant nous avons accès à l'interface d'administration de pgAdmin 4



Figure 30 : interface d'accueil pgadmin4

3.3 La réPLICATION

La réPLICATION dans PostgreSQL permet de créer et de maintenir des copies synchronisées d'une base de données PostgreSQL sur différents serveurs. Cela permet d'améliorer la disponibilité, la fiabilité et les performances du système.

PostgreSQL offre plusieurs méthodes de réPLICATION pour assurer la disponibilité et la redondance des données. Voici les deux principales méthodes de réPLICATION disponibles dans PostgreSQL :

- RéPLICATION continue : C'est la principal de PostgreSQL introduit dans la version 9.0. La réPLICATION en continu permet à un serveur de secours de rester plus à jour que ce qui est possible avec l'envoi de journaux basé sur des fichiers. Le serveur de secours se connecte au serveur principal, qui diffuse les enregistrements WAL vers le serveur de secours au fur et à mesure de leur génération, sans attendre que le fichier WAL soit rempli. PostgreSQL prend en charge deux modes de réPLICATION en continu : le mode asynchrone et le mode synchrone.^[9]
- RéPLICATION logique : Elle est un moyen de copier des données d'une base de données à une

autre de manière plus flexible et personnalisable que la réPLICATION physique. Au lieu de copier l'intégralité de la base de données, la réPLICATION logique se concentre sur la réPLICATION des modifications apportées à des lignes ou transactions individuelles.^[10]

3.3.1 Configuration de la réPLICATION

Nous allons faire la configuration de la réPLICATION logique sur Ubuntu.

- Le serveur de secours doit être de même architecture que le serveur principal (32 ou 64 bits)
- Nous allons créer deux machines virtuelles (**SM** et **SE**) comportant le même système d'exploitation via Oracle Virtual Box.

SM : i5 2 cœurs, 4gb Ram, Mémoire 50gb, Adresse IP : **192.168.1.172**

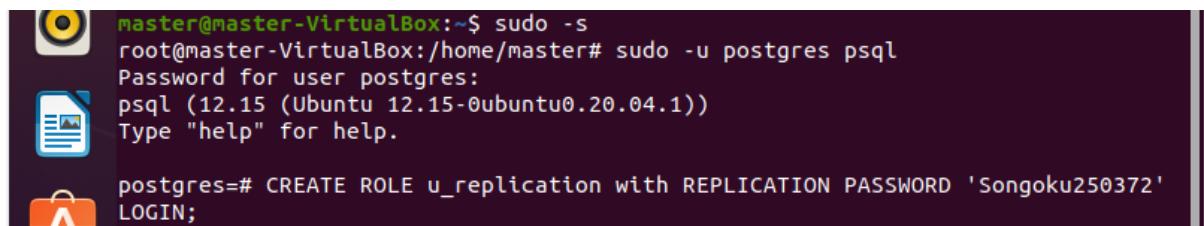
SE : i5 2 cœurs, 4gb Ram, Mémoire 50gb, Adresse IP : **192.168.1.186**

3.3.1.1 Configuration du serveur maître

Nous allons ouvrir un terminal et configurer le serveur maître en suivant les étapes suivantes :

-Création d'un rôle de réPLICATION sur le serveur

Nous allons créer un rôle de réPLICATION via la commande **CREATE ROLE**.



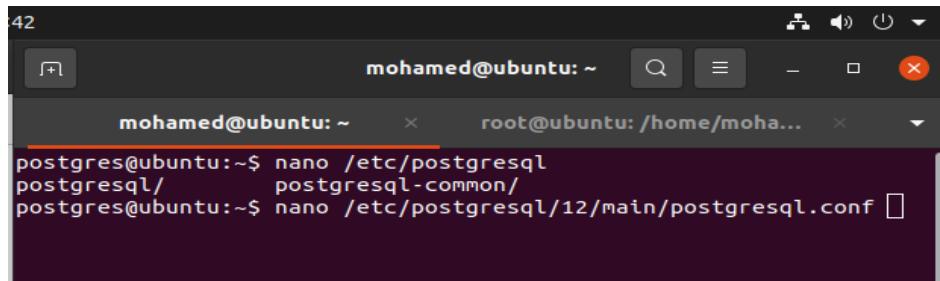
```
master@master-VirtualBox:~$ sudo -s
root@master-VirtualBox:/home/master# sudo -u postgres psql
Password for user postgres:
psql (12.15 (Ubuntu 12.15-0ubuntu0.20.04.1))
Type "help" for help.

postgres=# CREATE ROLE u_repliCATION WITH REPLICATION PASSWORD 'Songoku250372'
LOGIN;
```

Figure 31 : création du rôle de replication

Nous avons créé un utilisateur **u_repliCATION** avec un privilège de réPLICATION.

-Editons le fichier postgresql.conf avec la commande nano

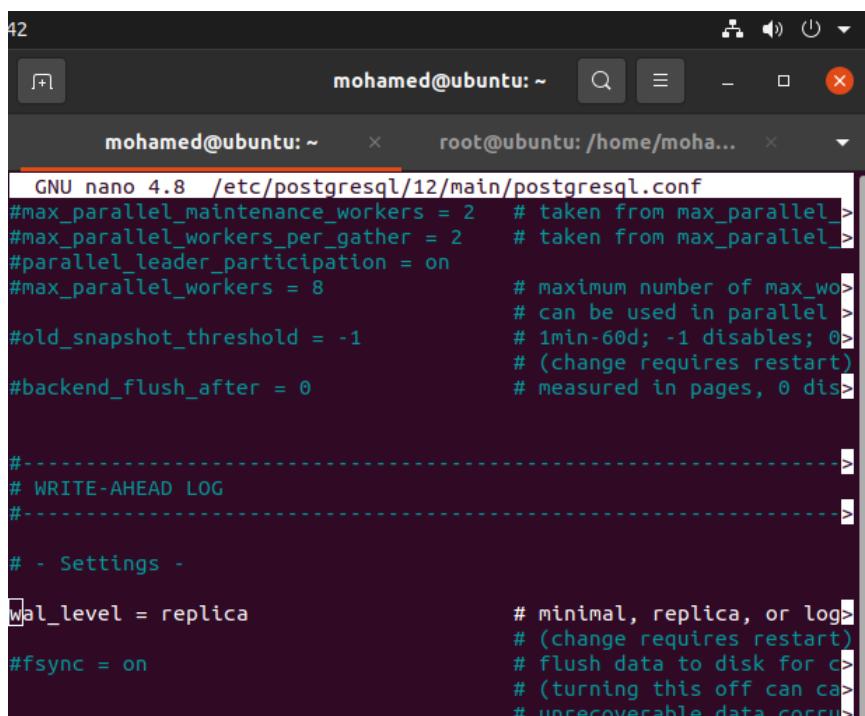


```
mohamed@ubuntu:~$ nano /etc/postgresql/12/main/postgresql.conf
postgres@ubuntu:~$ nano /etc/postgresql/12/main/postgresql.conf
```

Figure 32 : chemin de postgresql.conf

Et modifions les paramètres suivants :

- **wal_level:** Vous pouvez utiliser ce paramètre pour activer la réPLICATION PostgreSQL en continu, avec des valeurs possibles telles que minimal, replica, ou logical.



```
GNU nano 4.8 /etc/postgresql/12/main/postgresql.conf
#max_parallel_maintenance_workers = 2      # taken from max_parallel_>
#max_parallel_workers_per_gather = 2        # taken from max_parallel_>
#parallel_leader_participation = on         # maximum number of max_wo>
#max_parallel_workers = 8                   # can be used in parallel >
#old_snapshot_threshold = -1                # 1min-60d; -1 disables; 0>
#backend_flush_after = 0                  # measured in pages, 0 dis>

#-----#
# WRITE-AHEAD LOG
#-----#

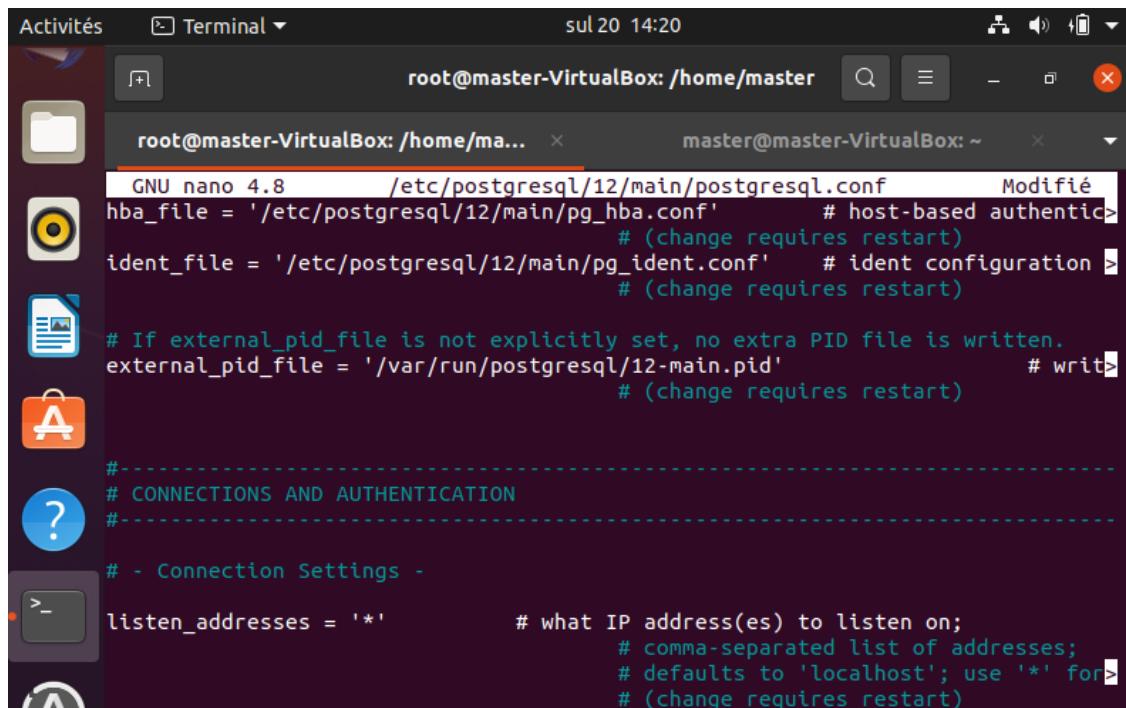
# - Settings -

wal_level = replica                      # minimal, replica, or log>
# (change requires restart)
#fsync = on                                # flush data to disk for c>
# (turning this off can cause data loss)
#unrecoverable_data_c...>
```

Figure 33 : activation de la réPLICATION

- **listen_addresses** : spécifie les adresses IP ou les interfaces réseau sur lesquelles le serveur de base de données écoute les connexions entrantes.

Changeons ce paramètre pour qu'elle écoute toutes les connexions entrantes en mettant **listen_addresses='*'**



```

root@master-VirtualBox: /home/master# nano /etc/postgresql/12/main/postgresql.conf
hba_file = '/etc/postgresql/12/main/pg_hba.conf'      # host-based authentication
                                                # (change requires restart)
ident_file = '/etc/postgresql/12/main/pg_ident.conf'  # ident configuration
                                                # (change requires restart)

# If external_pid_file is not explicitly set, no extra PID file is written.
external_pid_file = '/var/run/postgresql/12-main.pid'    # writes
                                                # (change requires restart)

#-----#
# CONNECTIONS AND AUTHENTICATION
#-----#
# - Connection Settings -
listen_addresses = '*'          # what IP address(es) to listen on;
                                # comma-separated list of addresses;
                                # defaults to 'localhost'; use '*' for
                                # (change requires restart)

```

Figure 34 : paramètre *listen_addresses*

Puis sauvegarder avec CTRL + S et fermer avec CTRL + X

-Éditer le fichier pg_hba.conf avec la commande nano

nano /etc/postgresql/<version>/main/pg_hba.conf

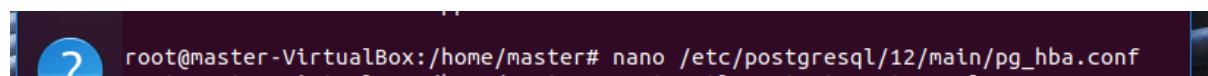


Figure 35 : chemin de pg_hba.conf

Et ajouter en fin de fichier le format suivant :

host replication rep_user IPaddress md5

- "host" : C'est le type d'hôte pour lequel vous configurez les règles d'authentification.
- "replication" : C'est le nom de la base de données pour laquelle vous spécifiez les règles d'authentification.
- "rep_user" : C'est le nom d'utilisateur (ou rôle) que vous utilisez pour la réPLICATION. Vous devrez remplacer "rep_user" par le nom réel de l'utilisateur utilisé pour la réPLICATION.
- "IPaddress" : C'est l'adresse IP du serveur depuis lequel vous autorisez la réPLICATION. Vous devrez remplacer "IPaddress" par l'adresse IP réelle du serveur.

- "md5" : C'est la méthode d'authentification utilisée pour vérifier le mot de passe de l'utilisateur. Lorsque vous utilisez "md5", PostgreSQL stocke les mots de passe chiffrés en utilisant l'algorithme MD5.

```
# If you want to allow non-local connections, you need to add more
# "host" records. In that case you will also need to make PostgreSQL
# listen on a non-local interface via the listen_addresses
# configuration parameter, or via the -i or -h command line switches.

# DO NOT DISABLE!
# If you change this first entry you will need to make sure that the
# database superuser can access the database using some other method.
# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket
local    all            postgres                      md5

# TYPE   DATABASE        USER        ADDRESS             METHOD
# "local" is for Unix domain socket connections only
local    all            all                      md5
# IPv4 local connections:
host     all            all          127.0.0.1/32      md5
# IPv6 local connections:
host     all            all          ::1/128           md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
local    replication   all                      md5
host    replication   all          127.0.0.1/32      md5
host    replication   all          ::1/128           md5
host    replication   u_replication  192.168.1.186/24  md5
```

Figure 36 : configuration de connexion du SE sur le SM

La dernière ligne permettra au serveur esclave ayant l'adresse IP 192.168.1.186 de se connecter à la base de données de réPLICATION avec l'identifiant u_replication en utilisant la méthode d'authentification MD5.

-Après configuration, il est nécessaire de redémarrer le service pour appliquer les modifications avec la commande en tant que root :

sudo systemctl restart <service>

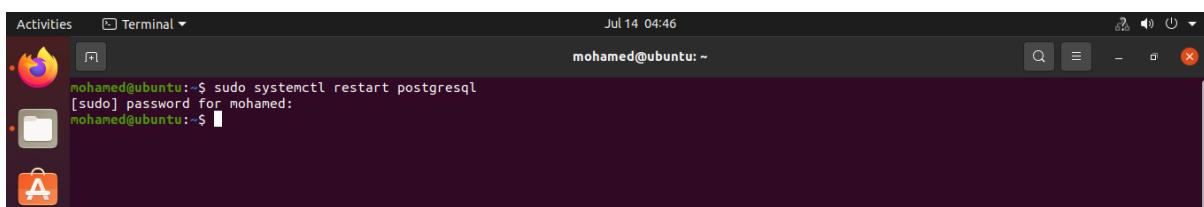
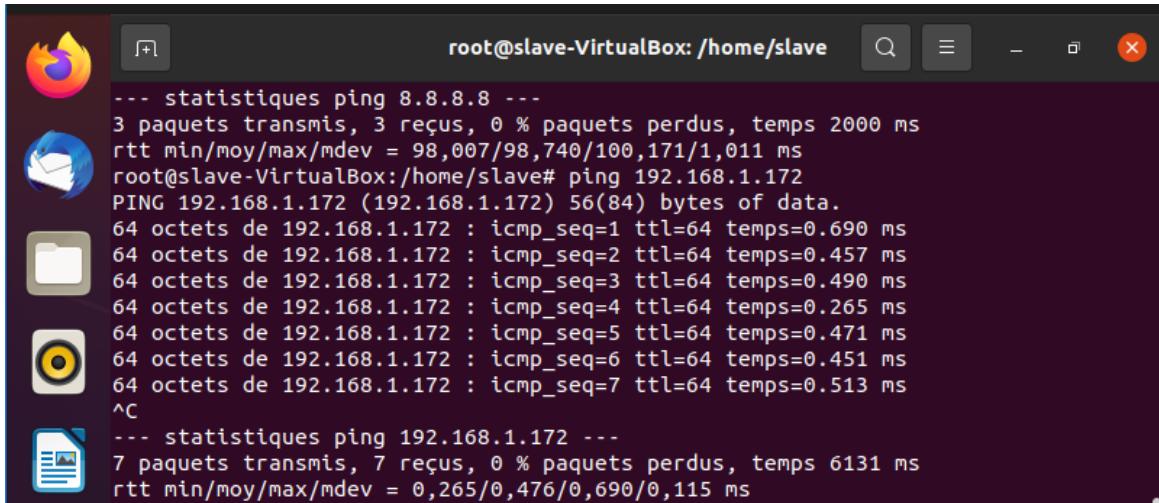


Figure 37 : redémarrage du service PostgreSQL

3.3.1.2 Configuration du serveur esclave

Avant de continuer, vérifions si les deux serveurs parviennent à communiquer avec la commande **ping** depuis le serveur esclave vers le serveur maître.



```
root@slave-VirtualBox: /home/slave
--- statistiques ping 8.8.8.8 ---
3 paquets transmis, 3 reçus, 0 % paquets perdus, temps 2000 ms
rtt min/moy/max/mdev = 98,007/98,740/100,171/1,011 ms
root@slave-VirtualBox:/home/slave# ping 192.168.1.172
PING 192.168.1.172 (192.168.1.172) 56(84) bytes of data.
64 octets de 192.168.1.172 : icmp_seq=1 ttl=64 temps=0.690 ms
64 octets de 192.168.1.172 : icmp_seq=2 ttl=64 temps=0.457 ms
64 octets de 192.168.1.172 : icmp_seq=3 ttl=64 temps=0.490 ms
64 octets de 192.168.1.172 : icmp_seq=4 ttl=64 temps=0.265 ms
64 octets de 192.168.1.172 : icmp_seq=5 ttl=64 temps=0.471 ms
64 octets de 192.168.1.172 : icmp_seq=6 ttl=64 temps=0.451 ms
64 octets de 192.168.1.172 : icmp_seq=7 ttl=64 temps=0.513 ms
^C
--- statistiques ping 192.168.1.172 ---
7 paquets transmis, 7 reçus, 0 % paquets perdus, temps 6131 ms
rtt min/moy/max/mdev = 0,265/0,476/0,690/0,115 ms
```

Figure 38 : test de communication entre le SM et le SE

On constate que les deux serveurs peuvent communiquer.

-Arrêtons le service PostgreSQL, puis connectons-nous en tant que postgres et faisons une sauvegarde du main directory.



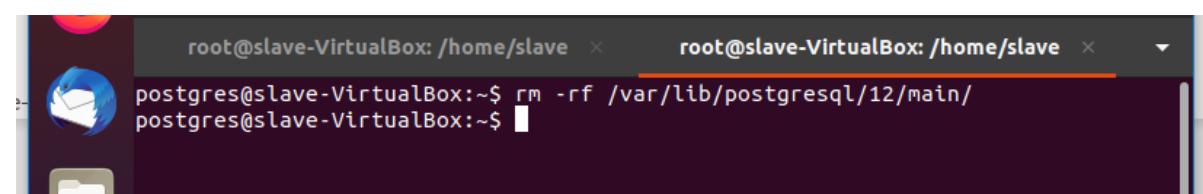
```
root@slave-VirtualBox:/home/slave# systemctl stop postgresql
root@slave-VirtualBox:/home/slave#
```



```
root@slave-VirtualBox:/home/slave# su - postgres
postgres@slave-VirtualBox:~$ cp -R /var/lib/postgresql/
12/.psql_history
postgres@slave-VirtualBox:~$ cp -R /var/lib/postgresql/12/main/ /var/lib/postgr
esql/
12/.psql_history
postgres@slave-VirtualBox:~$ cp -R /var/lib/postgresql/12/main/ /var/lib/postgr
esql/12/main_old/
postgres@slave-VirtualBox:~$
```

Figure 39 : sauvegarde du main directory

-ensuite supprimons le main original



```
root@slave-VirtualBox: /home/slave ×      root@slave-VirtualBox: /home/slave ×
postgres@slave-VirtualBox:~$ rm -rf /var/lib/postgresql/12/main/
postgres@slave-VirtualBox:~$
```

Figure 40 : suppression du main

-Faire une sauvegarde de base pour amorcer le serveur esclave

Nous utiliserons la commande pg_basebackup pour créer une sauvegarde de base d'un cluster de bases de données PostgreSQL.

Dans la commande suivante, l'option :

- **-h** – spécifie l'hôte qui est le serveur maître.
- **-D** – spécifie le répertoire de données.
- **-U** – spécifie l'utilisateur de la connexion.
- **-P** : active les rapports d'avancement.
- **-v** – active le mode détaillé.
- **-R** – permet la création d'une configuration de récupération : crée un fichier standby.signal et ajoute les paramètres de connexion à postgresql.auto.conf sous le répertoire de données.
- **-X** – utilisé pour inclure les fichiers journaux à écriture anticipée requis (fichiers WAL) dans la sauvegarde. Une valeur de flux signifie diffuser le WAL pendant que la sauvegarde est créée.
- **-C** – permet la création d'un emplacement de réPLICATION nommé par l'option **-S** avant de démarrer la sauvegarde.
- **-S** – spécifie le nom de l'emplacement de réPLICATION.^[11]

```
postgres@slave-VirtualBox:~$ pg_basebackup -h 192.168.1.172 -D /var/lib/postgresql/12/main/ -U u_replication -P -v -R -X stream -C -S slaveslot1
Password:
pg_basebackup: initiating base backup, waiting for checkpoint to complete
pg_basebackup: checkpoint completed
pg_basebackup: write-ahead log start point: 0/2000028 on timeline 1
pg_basebackup: starting background WAL receiver
pg_basebackup: created replication slot "slaveslot1"
24650/24650 kB (100%), 1/1 tablespace
pg_basebackup: write-ahead log end point: 0/2000138
pg_basebackup: waiting for background process to finish streaming ...
pg_basebackup: syncing data to disk ...
pg_basebackup: base backup completed
postgres@slave-VirtualBox:~$ ls /var/
backups/ crash/ local/ log/ metrics/ run/ spool/
cache/ lib/ lock/ mail/ opt/ snap/ tmp/
postgres@slave-VirtualBox:~$ ls /var/lib/postgresql/
12/ .pgsql_history
postgres@slave-VirtualBox:~$ ls /var/lib/postgresql/12/main
main/ main_old/
```

Figure 41 : sauvegarde de base du main SM

A la dernière ligne nous constatons qu'il existe un nouveau fichier main provenant du maître.

-Lorsque le processus de sauvegarde est terminé, le nouveau répertoire de données sur le SE devrait ressembler à celui de la capture d'écran. Un standby.signal est créé et les paramètres de connexion sont ajoutés à postgresql.auto.conf. Vous pouvez lister son contenu à l'aide de la commande ls.

```

postgres@slave-VirtualBox:~$ ls -ltrh /var/lib/postgresql/12/main
main/    main_old/
postgres@slave-VirtualBox:~$ ls -ltrh /var/lib/postgresql/12/main/
total 80K
drwx----- 2 postgres postgres 4,0K sul 20 15:42 pg_twophase
drwx----- 2 postgres postgres 4,0K sul 20 15:42 pg_tblspc
drwx----- 2 postgres postgres 4,0K sul 20 15:42 pg_stat_tmp
drwx----- 2 postgres postgres 4,0K sul 20 15:42 pg_stat
drwx----- 2 postgres postgres 4,0K sul 20 15:42 pg_replslot
drwx----- 4 postgres postgres 4,0K sul 20 15:42 pg_multixact
drwx----- 2 postgres postgres 4,0K sul 20 15:42 pg_dynshmem
-rw----- 1 postgres postgres 224 sul 20 15:42 backup_label
drwx----- 3 postgres postgres 4,0K sul 20 15:42 pg_wal
drwx----- 2 postgres postgres 4,0K sul 20 15:42 pg_xact
drwx----- 2 postgres postgres 4,0K sul 20 15:42 pg_notify
drwx----- 5 postgres postgres 4,0K sul 20 15:42 base
-rw----- 1 postgres postgres 3 sul 20 15:42 PG_VERSION
drwx----- 2 postgres postgres 4,0K sul 20 15:42 pg_subtrans
drwx----- 2 postgres postgres 4,0K sul 20 15:42 pg_snapshots
drwx----- 2 postgres postgres 4,0K sul 20 15:42 pg_serial
drwx----- 4 postgres postgres 4,0K sul 20 15:42 pg_logical
drwx----- 2 postgres postgres 4,0K sul 20 15:42 pg_commit_ts
drwx----- 2 postgres postgres 4,0K sul 20 15:42 global
-rw----- 1 postgres postgres 0 sul 20 15:42 standby.signal
-rw----- 1 postgres postgres 308 sul 20 15:42 postgresql.auto.conf
postgres@slave-VirtualBox:~$ 
```

Figure 42 : contenu du nouveau main

-Démarrer le service postgresql avec la commande :

systemctl start postgresql

et vérifions avec

systemctl status postgresql

```

postgres@slave-VirtualBox:~$ systemctl start postgresql
postgres@slave-VirtualBox:~$ systemctl status postgresql
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; enabled; vendor p>
   Active: active (exited) since Thu 2023-07-20 16:03:06 GMT; 25s ago
     Process: 8192 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
    Main PID: 8192 (code=exited, status=0/SUCCESS)
      Tasks: 0
     CGroup: /system.slice/postgresql.service

```

Figure 43 : statut de PostgreSQL

-De retour sur le serveur maître, nous allons vérifier si le slot de réPLICATION appéLé slaveslot1 est présent lorsque nous ouvrons la vue pg_replication_slots comme suit.

```
root@master-VirtualBox:/home/master# sudo -u postgres psql
Password for user postgres:
psql (12.15 (Ubuntu 12.15-0ubuntu0.20.04.1))
Type "help" for help.

postgres=# SELECT * FROM pg_replication_slots
postgres-# ;
```

pid	usesysid	username	application_name	client_addr	client_hostname	client_port	backend_start	backend_xmin	state	sent_lsn	wrt
9396	16384	u_repliCation	12/main	192.168.1.186		0	45344	2023-07-20 16:02:59.887465+00	streaming	0/3000148	0/3
00148	0/3000148	0/3000148				0	async	2023-07-20 16:12:06.703451+00			

(1 row)

Figure 44 : information détaillé de la réPLICATION

3.3.2 Test de la réPLICATION

Nous allons à présent faire les tests de la réPLICATION en continue.

-Sur le serveur esclave essayons de faire une transaction

```
postgres@slave-VirtualBox:~$ psql
Password for user postgres:
psql (12.15 (Ubuntu 12.15-0ubuntu0.20.04.1))
Type "help" for help.

postgres=# create database slave1;
ERROR:  cannot execute CREATE DATABASE in a read-only transaction
postgres=#
```

Figure 45 : Essai de transaction sur SE

On constate que maintenant le serveur esclave est en lecture seul a défaut de configurer le basculement automatique

-Sur le serveur maître créons une base de données nommée test et vérifions s'il est répliqué sur le serveur esclave.

- Création de la base de donne

```
postgres=# create database test;
CREATE DATABASE
postgres=#

```

Figure 46 : création de base de donnée 'test' sur le SM

- Vérification sur l'esclave

```
postgres=# select datname from pg_database;
datname
-----
postgres
test
template1
template0
(4 rows)

postgres=#
```

Figure 47 : base de donnée 'test' bien répliquée sur le SE

Nous constatons que la base de données test a été bien répliquée.

-Maintenant sur le master créons une table dans la base de données 'test', insérons-y des données et vérifions s'ils ont été répliqués sur le serveur esclave.

- **Création de la table sur le master**

Pour se connecter sur une base de données postgresql

`psql -U nom_utilisateur -d nom_base_de_données`

ou

`sudo -u nom_utilisateur psql nom_base_de_données`

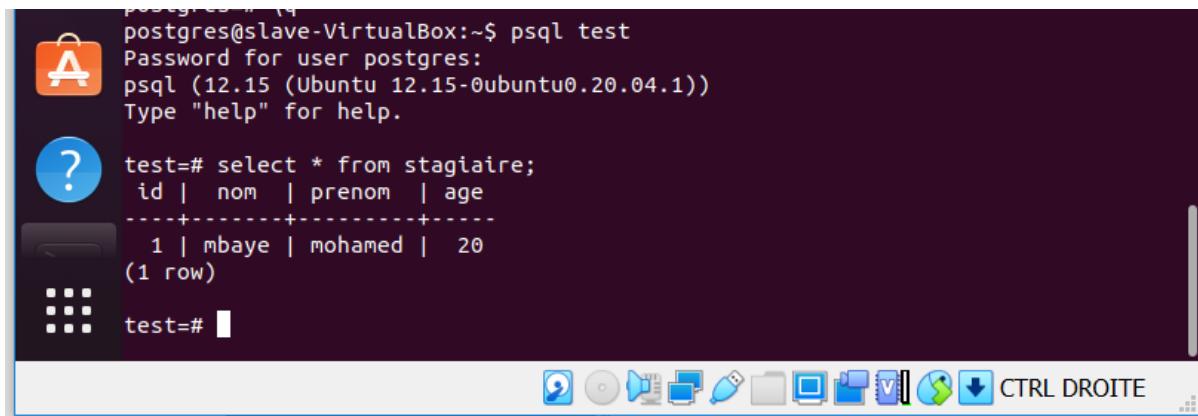
```
root@master-VirtualBox:/home/master# sudo -u postgres psql test
Password for user postgres:
psql (Ubuntu 12.15-0ubuntu0.20.04.1)
Type "help" for help.

test=# CREATE TABLE stagiaire(id serial PRIMARY KEY,nom VARCHAR(30),prenom VARCHAR(30),age int);
CREATE TABLE
test=# INSERT INTO stagiaire(nom,prenom,age) VALUES ('mbaye','mohamed',20);
INSERT 0 1
test=# SELECT * FROM stagiaire;
 id | nom    | prenom | age
----+-----+-----+
  1 | mbaye | mohamed | 20
(1 row)

test=#
```

Figure 48 :création de la table stagiaire dans la base de donnée 'test' sur le SM

- **Vérification sur le serveur esclave**



A screenshot of a terminal window titled 'postgres:n~' on a 'slave-VirtualBox'. The window shows the PostgreSQL command-line interface (psql) connected to a database named 'test'. The user 'postgres' is prompted for a password. Once connected, the command 'select * from stagiaire;' is run, displaying a single row of data: 'id | nom | prenom | age' with values '1 | mbaye | mohamed | 20'. The terminal window has a dark background and includes standard Linux desktop icons in its dock at the bottom.

```
postgres:n~ postgres@slave-VirtualBox:~$ psql test
Password for user postgres:
psql (12.15 (Ubuntu 12.15-0ubuntu0.20.04.1))
Type "help" for help.

test=# select * from stagiaire;
 id | nom | prenom | age
----+-----+-----+-----
  1 | mbaye | mohamed | 20
(1 row)

test=#
```

Figure 49 : table stagiaire bien répliqué sur le SE

Nous constatons que les données ont été bien répliquées sur le serveur esclave.

3.3.3 Conclusion

Ce chapitre nous a permis aborder en détail l'implémentation de PostgreSQL dans notre environnement Ubuntu, ainsi que la configuration de la réPLICATION. Nous avons également présenté les choix d'outils et de technologies que nous avons utilisés pour mener à bien cette mise en œuvre.

CONCLUSION GENERAL

Dans le cadre de ce mémoire, durant mes 60 jours de stage au sein de la SENELEC notre objectif principal était de répondre aux besoins spécifiques de SENELEC en matière de gestion de bases de données PostgreSQL. Pour atteindre ces objectifs, nous avons élaboré une stratégie complète, couvrant chaque étape, de l'analyse des besoins initiaux à la mise en place de la réPLICATION.

Au fil de ce mémoire, nous avons pu tirer plusieurs enseignements essentiels. Tout d'abord, nous avons constaté que PostgreSQL est une solution puissante, souple et robuste pour la gestion de bases de données. Sa conformité aux normes SQL, sa communauté active et sa capacité à évoluer avec les besoins de l'entreprise en font un choix judicieux.

En somme, ce projet est un témoignage de l'engagement de SENELEC envers l'innovation et la qualité de service, et nous sommes convaincus qu'il continuera à contribuer de manière significative à la réussite de l'entreprise dans les années à venir.

WEBOGRAPHIE

[1]https://fr.wikipedia.org/wiki/Soci%C3%A9t%C3%A9_nationale_d%27%C3%A9lectricit%C3%A9_du_S%C3%A9n%C3%A9gal (dernière date de consultation: le 15 Septembre 2023)

[2]https://fr.wikipedia.org/wiki/Soci%C3%A9t%C3%A9_nationale_d%27%C3%A9lectricit%C3%A9_du_S%C3%A9n%C3%A9gal (dernière date de consultation: le 15 Septembre 2023)

[3] <https://www.senelec.sn/direction-des-systemes-dinformation-dsi> (dernière date de consultation: le 15 Septembre 2023)

[4] [https://fr.wikipedia.org/wiki/Ubuntu_\(syst%C3%A8me_d%27exploitation\)](https://fr.wikipedia.org/wiki/Ubuntu_(syst%C3%A8me_d%27exploitation)) (dernière date de consultation: le 16 Septembre 2023)

[5] <https://www.virtualbox.org/> (dernière date de consultation: le 16 Septembre 2023)

[6]: <https://www.postgresql.org/> (dernière date de consultation: le 07 Juillet 2023)

[7] <https://kinsta.com/fr/base-de-connaissances/qu-est-postgresql/> (dernière date de consultation: le 09 Juillet 2023)

[8] <https://kinsta.com/fr/base-de-connaissances/qu-est-postgresql/> (dernière date de consultation: le 09 Juillet 2023)

[9] <https://www.enterprisedb.com/blog/how-set-streaming-replication-keep-your-postgresql-database-performant-and-durable>(dernière date de consultation: le 12 Juillet 2023)

[10] https://www.percona.com/blog/how-to-start-logical-replication-in-postgresql-for-specific-tables-based-on-a-pg_dump/#:~:text=To%20set%20up%20logical%20replication,starts%20an%20application%20worker%20process. (dernière date de consultation: le 15 Juillet 2023)

[11] <https://kinsta.com/blog/postgresql-replication/> (dernière date de consultation: le 22 Juillet 2023)