

**Configuration d’un serveur de bases de données**

Projet gsb.org

DUMAS Lucie

Table des matières

[Qu’est-ce que GPG 3](#_Toc162435592)

[Configuration de GPG 4](#_Toc162435593)

[La commande sudo 13](#_Toc162435594)

[Différence entre les commandes su, su- et sudo 13](#_Toc162435595)

[Configuration de la commande sudo 13](#_Toc162435596)

# Qu’est-ce que MariaDB

MariaDB est un système de gestion de base de données open source qui est une alternative compatible avec MySQL.

Il offre un stockage et une récupération efficace des données, la prise en charge du langage SQL, une grande extensibilité et une communauté active de développeurs.

MariaDB est utilisé pour stocker, organiser et récupérer des données dans de nombreuses applications, des sites web aux applications d'entreprise.Configuration de GPG

# Installation de MariaDB

## Installation

Dans un premier temps, nous commençons par cloner notre machine virtuelle template pour créer une nouvelle machine (priv-db1 ou priv-db2). Nous changeons le nom d'hôte ainsi que l'adresse IP (10.31.177.33 pour priv-db1 et 10.31.178.33) :

# Change le nom d'hôte à l'aide d'une commande

hostnamectl set-hostname priv-db1 # Ou priv-db2

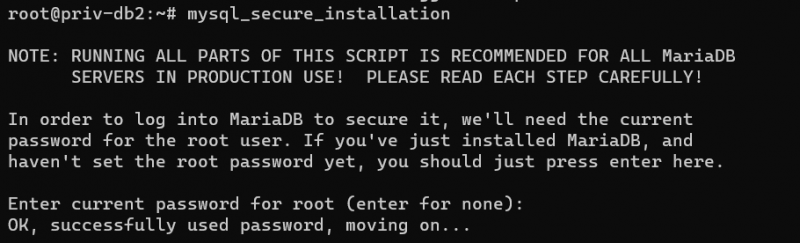
# Change le nom d'hôte directement depuis le fichier de configuration

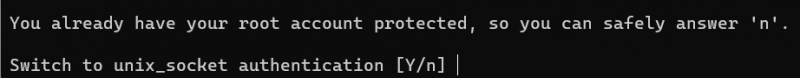
nano /etc/hosts

Nous modifions dans le fichier /etc/hosts le nom d’hôte de la machine et nous nous déconnectons pour actualiser l’affichage du nom.

Nous pouvons maintenant procéder à l’installation sécurisée de MariaDB. Nous entrons donc la commande suivante :

mysql\_secure\_installation

 En ce qui concerne l’installation de MariaDB, nous laisserons les paramètres par défaut de l’outil. Nous changerons cependant le mot de passe du compte root :



## Vérification de la configuration par défaut

## Conformité du cahier des charges

# dzd

Dans un premier temps, nous allons télécharger sur toutes nos machines le paquet GPG :

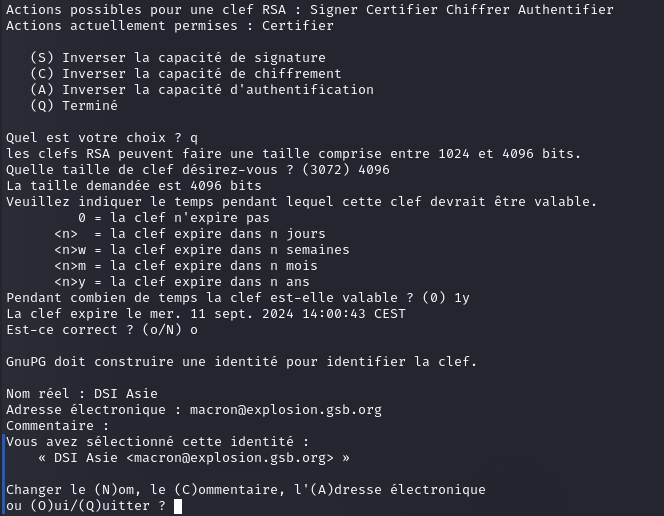
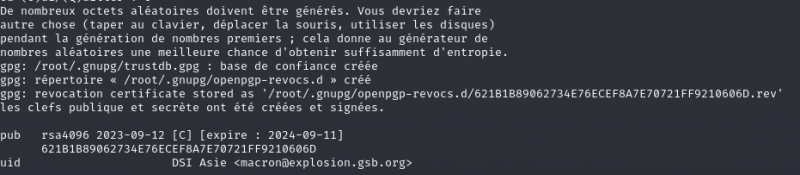
apt update && apt upgrade

apt install gpg

Sur notre routeur, nous allons générer une paire de clés publique et privée en utilisant l’outil GPG à l’aide de la commande suivante :

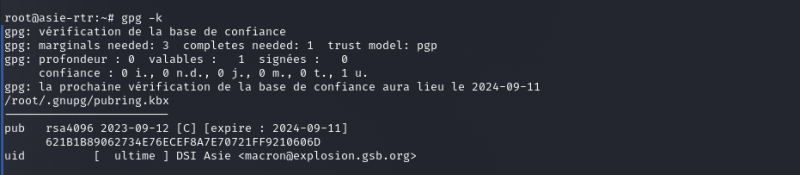
gpg --full-generate-key --expert

 Nous créons une clé de certification. Pour cela, nous choisissons le type de clé RSA (8) et nous enlevons les options de signature et de chiffrement en sélectionnant successivement (S) puis (C) et enfin (Q) pour quitter. Nous lui choisissons une taille de 4096 bits puis une durée de validité de 1 an. Nous pouvons maintenant générer la clé.



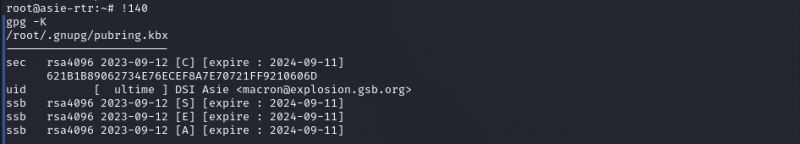
Nous vérifions que la clé publique soit bien générée :

gpg -k



Nous vérifions également que la clé privée soit bien générée :

gpg -K



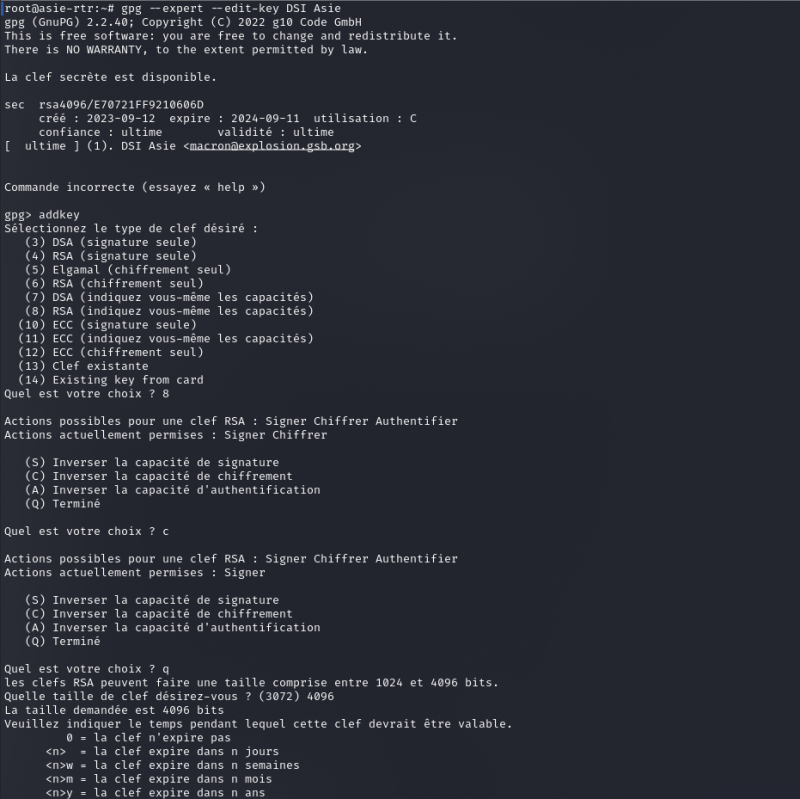
Nous allons maintenant créer une sous-clé de signature :

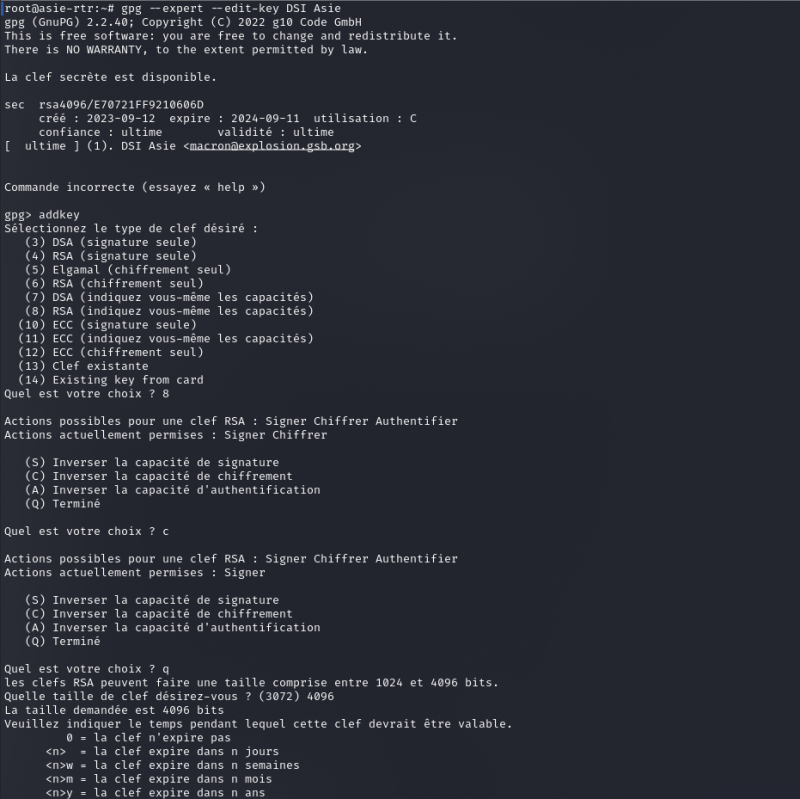
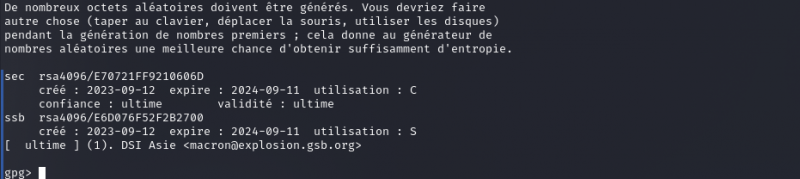
# Pour cette commande il faut préciser --edit-key pour éditer la clé puis indiquer le nom de la clé que nous souhaitons modifier.

gpg --expert --edit-key DSI Asie

addkey

Nous sélectionnons le type de clé RSA (8), puis nous enlevons l’option de chiffrement (C) et enfin (Q) pour quitter. Nous lui choisissons une taille de 4096 bits puis une durée de validité de 1 an. Nous pouvons générer la sous-clé.

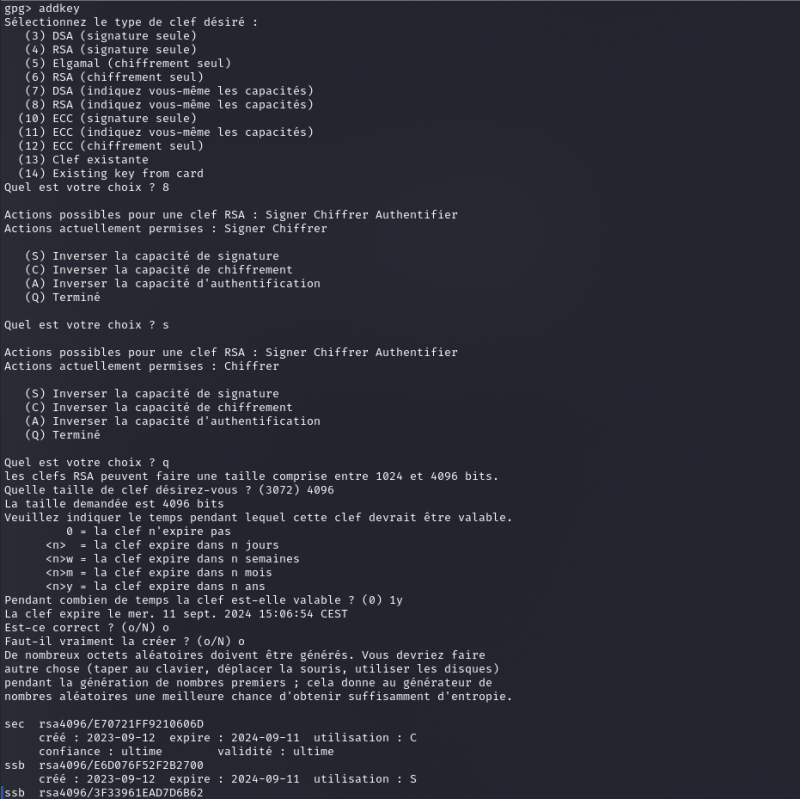




Nous créons maintenant une sous-clé de chiffrement :

addkey

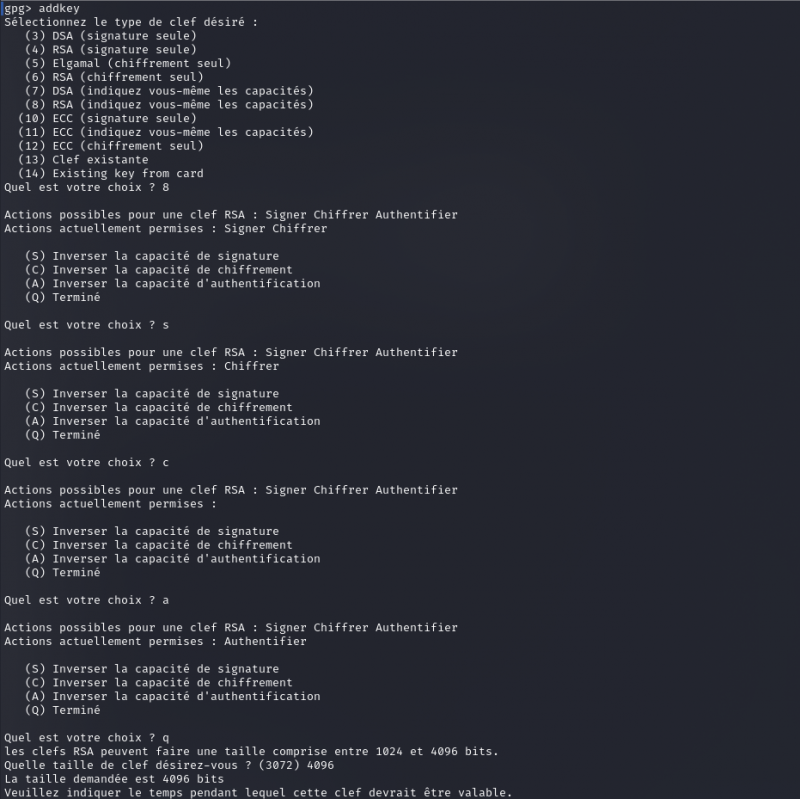
Nous sélectionnons le type de clé RSA (8), puis nous enlevons l’options de signature (S) et enfin (Q) pour quitter. Nous lui choisissons une taille de 4096 bits puis une durée de validité de 1 an. Nous pouvons générer la sous-clé.



Nous créons une sous-clé d’authentification :

addkey

Nous sélectionnons le type de clé RSA (8), puis nous enlevons l’option de signature (S), de chiffrement (C), nous ajoutons l’option d’authentification (A) et enfin, nous quittons le menu de configuration (Q). Nous lui choisissons une taille de 4096 bits, puis une durée de validité de 1 an. Nous pouvons générer la sous-clé.



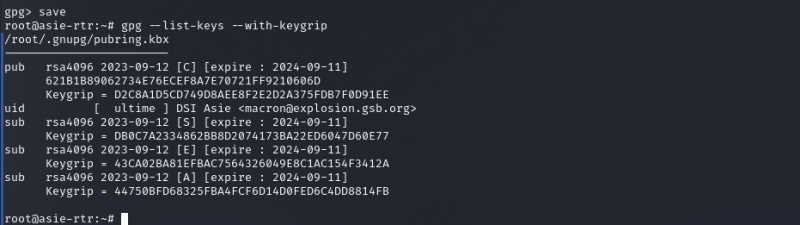
Nous sauvegardons les changements effectués :

save

# L'option --list-keys permet de lister les clé présente

# L'option --with-keygrip permet d'afficher le grip de la clé d'authentification.

gpg --list-keys --with-keygrip



Une fois nos clés créées, nous allons copier les clés publiques suivantes dans le fichier ~/.ssh/authorized\_keys afin que les machines possédant ces clés publiques puissent se connecter au serveur :

* la clé publique du routeur Asie
* la clé publique du routeur Monde

Sur le serveur, nous allons importer la clé publique du routeur ainsi que sa clé privé pour autoriser la connexion du serveur vers le routeur et du routeur vers le serveur. Nous importons la clé publique dans le fichier ~/.ssh/authorized\_keys.

Pour qu'une machine autorise une connexion SSH par clé GPG, nous devons importer la clé publique du routeur dans le fichier ~/.ssh/authorized\_keys. Pour qu'une machine puisse se connecter à une autre machine autorisant la clé publique du routeur, il faut que cette dernière soit en possession de la clé privée du routeur.

Etant donné que, par défaut, l'agent SSH ne reconnait pas les clés GPG, nous devons activer pour chaque machine (conteneurs et machines virtuelles inclus) l'agent GPG dans le fichier ~/.gnupg/gpg-agent.conf afin que ce dernier puisse prendre en charge les clés.

enable-ssh-support >> $HOME/.gnupg/gpg-agent.conf

Nous modifions également notre fichier ~/.bash\_profile pour y ajouter un script permettant d’échanger les sockets SSH et GPG afin que l’on puisse utiliser les clés GPG pour une connexion SSH.

nano ~/.bash\_profile

# Script d'activation de l'agent GPG

unset SSH\_AGENT\_PID

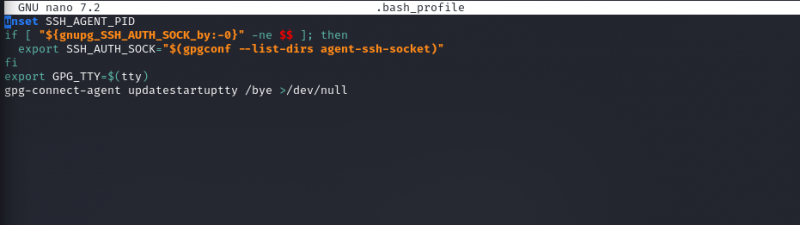
if [ "${gnupg\_SSH\_AUTH\_SOCK\_by:-0}" -ne $$ ]; then

export SSH\_AUTH\_SOCK="$(gpgconf --list-dirs agent-ssh-socket)"

fi

export GPG\_TTY=$(tty)

gpg-connect-agent updatestartuptty /bye >/dev/null



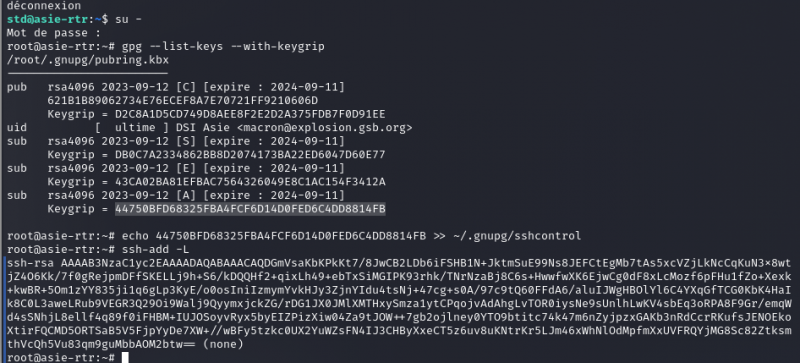
Pour que les machines puissent prendre en compte tous les changements, nous nous déconnectons du compte utilisateur et nous nous reconnectons. Nous pouvons afficher la liste des clés afin de vérifier que les changements aient bien été pris en compte.

# Afficher la liste des clés avec le grip de la clé d'authentification

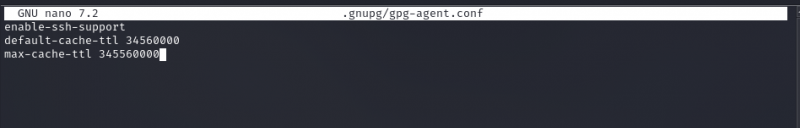
gpg --list-keys --with-grip

echo 44750BFD68325FBA4FCF6D14D0FED6C4DD8814FB >> ~/.gnupg/sshcontrol

ssh-add -L



Nous pouvons ensuite modifier le Time To Live de notre passphrase afin de ne pas avoir à la rentrer chaque jours. Pour cela, nous modifions le fichier  ~/.gnupg/gpg-agent.conf :



# La commande sudo

## Différence entre les commandes su, su- et sudo

La commande su permet de se connecter en tant que root en gardant les variables d’environnement de l’utilisateur précédent.

La commande su – permet de se connecter en tant que root en utilisant les variables d’environnement de l’utilisateur root, ce qui permet d’utiliser des commandes apparaissant comme introuvables pour les autres utilisateurs.

La commande sudo permet d’entrer une commande avec les privilèges administrateurs si l’utilisateur fait parti des groupes pouvant utiliser la commande sudo.

## Configuration de la commande sudo

Pour configurer la commande sudo, nous devons ajouter l’utilisateur std dans le groupe sudo :

usermod -a -G sudo std

Pour que les changements soient effectifs, nous devons nous déconnecter puis nous reconnecter à l’utilisateur std. Nous pouvons répéter cette manipulation sur chaque machine de notre réseau pour nous assurer que l’utilisateur std puisse utiliser la commande sudo.

Pour vérifier que l’utilisateur std soit dans le groupe sudo, nous utilisons la commande suivante :

cat /etc/group

 Nous pouvons voir la liste des groupes et des utilisateurs qui leurs sont associés :