Document d'installation d'un serveur Nexcloud distant de la base de donnée MySQL MariaDB avec différentes sécurités

Antoine Laguette & Juliette Bluem

14 avril 2022

Table des matières

Partie I : MariaDB

1 Installation MariaDB

Avant de commencer l'installation, quelques pré-requis :
Votre machine doit être à jour, vous devez avoir le statut root ou bien le paquet sudo d'installé et vous devez avoir installé les dépendances et packages nécessaires :

```
> su -
> apt update
> apt install curl
> sudo apt-get install curl software-properties-common dirmngr ca-certificates
apt-transport-https -y
```

Nous commençons l'installation par importer le réferentiel MariaDB :

```
> curl -LsS https://downloads.mariadb.com/MariaDB/mariadb_repo_setup | sudo bash -s --
--mariadb-server-version=10.7 --skip-maxscale --skip-tools
```

Puis par installer les packages serveur / client :

```
> apt install mariadb-server mariadb-client
```

Vous pouvez vérifier l'état du service MariaDB avec la commande :

```
> systemctl status mariadb
```

Si le service n'est pas actif, vous pouvez le lancer :

```
> systemctl start mariadb
```

(pour l'arrêter, remplacez "start" par "stop")

2 Utilisation MariaDB

Nous vous conseillons de lancer automatiquement le service MySQL au démarage de la marchine Debian.

```
> systemctl enable mariadb
```

Nous allons maintenant créer une vraie base de donnée et un utilisateur distant.
Lancez MariaDB puis créez la base :

```
> mariadb
>> CREATE DATABASE 'nomDB';
```

Créons maintenant un utilisateur distant, l'IP doit correspondre a l'adresse de la machine qui va se connecter :

```
>>CREATE USER pseudo@IP IDENTIFIED BY 'passwd';
```

Enfin, nous allons accorder tous les privilèges nécessaires à l'utilisateur sur notre base de données.

```
>>GRANT ALL PRIVILEGES ON nomDB.* TO pseudo@IP;
>>FLUSH nomDB
```

Notre base de donnée est désormais prête a accueillir notre application Nextcloud.

Partie II : Nextcloud

1 Installation d'apache

Dans un premier temps avant de pouvoir installer Nextcloud, il nous faut un accès à l'application par une page web. Ici nous allons choisir d'utiliser Apache mais il existe d'autre solution comme Nginx. Avant de commencer l'installation, on s'assure que notre système debian a son repo cache à jour avec la commande :

```
> sudo apt update & upgrade
```

On procède ensuite à l'installation d'Apache et son activation au démarrage du système :

```
> sudo apt install apache
> sudo systemctl enable apache
```

Nous allons désormais créer une configuration pour notre application Nextcloud. Pour cela nous allons devoir déposer la configuration dans les fichiers d'apache :

```
> sudo nano /etc/apache2/sites-available/nextcloud.conf
```

On ajoute ensuite la configuration qui suit :

```
<VirtualHost *:80>
     ServerAdmin admin@example.com
     DocumentRoot /var/www/html/nextcloud
     ServerName example.com
     ServerAlias www.example.com

     <Directory /var/www/html/nextcloud/>
          Options FollowSymlinks
          AllowOverride All
          Require all granted
     </Directory>

     ErrorLog ${APACHE_LOG_DIR}/error.log
     CustomLog ${APACHE_LOG_DIR}/access.log combined

     <Directory /var/www/html/nextcloud/>
            RewriteEngine on
            RewriteBase /
            RewriteCond %{REQUEST_FILENAME} !-f
            RewriteRule ^(.*) index.php [PT,L]
    </Directory>
</VirtualHost>
```

Avant de redémarrer notre Apache, nous allons activer notre configuration et désactiver l'ancienne :

```
> sudo a2dissite 000-default.conf
> sudo a2ensite nextcloud.conf

> sudo systemctl reload apache2
```

2 Installation de Nextcloud

Une fois Apache correctement configuré, nous allons pouvoir installer Nextcloud. Pour cela, rendez vous sur le site officiel Nextcloud pour récupérer le lien de téléchargement de la dernière version de l'application. Puis à l'aide du packet wget la télécharger sur notre machine :

```
> wget https://download.nextcloud.com/server/releases/nextcloud-XXX.zip
```

Il faut désormais décompresser notre archive dans le dossier root de Apache.

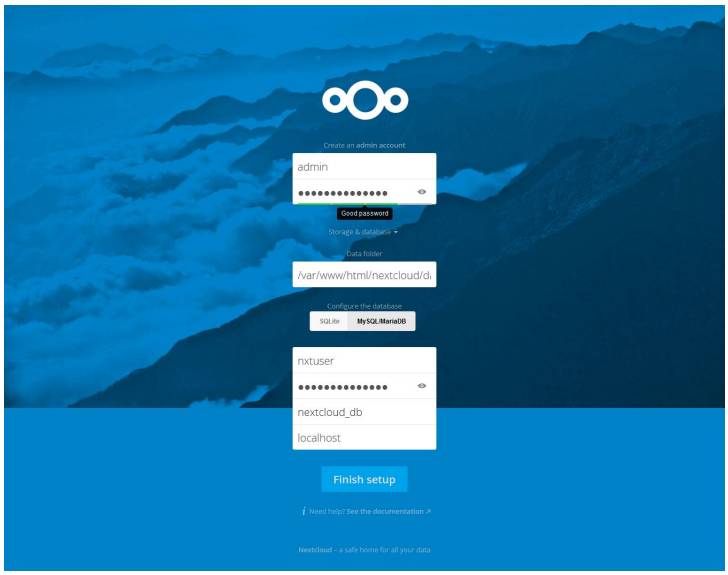
```
> unzip nextcloud-*.zip
> sudo mv nextcloud /var/www/html/
```

Ensuite il ne faut pas oublier de transférer les droits du dossier a l'utilisateur Apache

```
> sudo chown -R www-data:www-data /var/www/html/nextcloud/
```

3 Configuration de Nextcloud

Désormais rendez vous sur http ://Votreip pour accéder a l'interface Nextcloud.



Sur cette interface, vous allez dans les deux premières cases entrer le nom et le mot de passe que vous souhaitez de votre utilisateur Nextcloud. Il n'y a normalement rien a modifier sur la case centrale, si vous avez changer le répertoire des données nextcloud, indiquez le nouveau chemin. Sinon laissez par défaut.
La dernière partie a remplir concerne la connexion avec la base de donnée MariaDB. Rentrer le nom et le mot de passe de votre utilisateur MariaDB que vous avez crée pour Nextcloud, rentrez ensuite le nom de la base de donnée. Enfin comme notre serveur de données n'est pas sur la même machine, nous allons devoir entrer l'ip de la machine contenant MariaDB afin de s'y connecter. A la suite de l'ip il est important de préciser le port de connexion, par défault c'est le port 3306.
Cliquez sur finish setup pour lancer l'initialisation de Nextcloud, si vous n'avez pas fait d'erreur vous devriez atterrir sur l'interface administrateur de nextcloud.

Partie III : Les flux

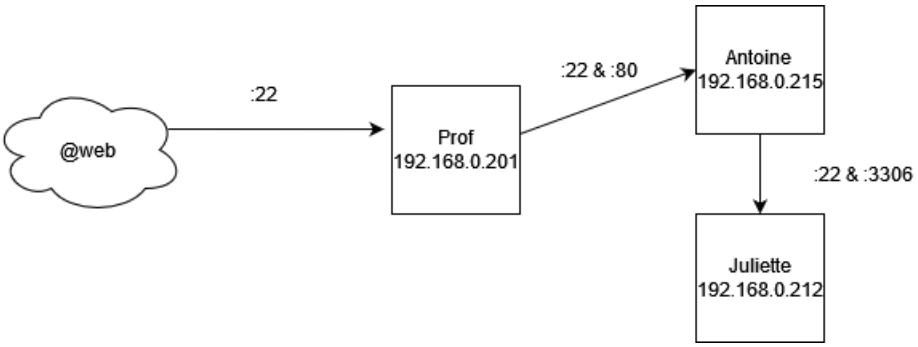


FIGURE 1 – Schémas des flux

Partie IV : Les sécurités

1 Configuration du Pare-feu iptables

Pour sécuriser notre application, nous avons configurer les pares-feu avec des règles très stricte.
Pour la machine Nextcloud, nous n'avons autoriser que la machine d'accès (machine professeur) a venir sur les ports 80 pour l'accès a l'application et au port 22 pour l'accès ssh. La machine ne peut pas initier de connections avec l'extérieur d'elle même non plus.
Pour la machine MariaDB les règles sont encore plus stricte afin de protéger au mieux notre base de donnée. Seul la machine Nextcloud est autoriser a venir parler sur les ports 3306 pour accéder a la base de donnée et au port 22 pour venir en ssh. Cela veut dire que si l'on souhaite accéder en ssh a MariaDB il faut d'abord passer par la machine Nextcloud. Ce qui implique de connaitre les accès.

2 Ajout d'outils comme fail2ban

Afin de renforcer notre sécurité on peut ajouter des outils comme fail2ban afin de limiter les tentatives d'intrusions par bruteforce.

3 Règles de pare-feu Nextcloud

```
#!/usr/sbin/nft -f

flush ruleset

table ip antoinenc {
    chain input {
        type filter hook input priority 0; policy drop;

        # Accept all localhost traffic
        iif lo accept
        ip saddr 127.0.0.0/8 counter drop

        # Drop invalid connections
        ip protocol tcp ct state invalid counter drop
        ip protocol udp ct state invalid counter drop
        ip protocol icmp ct state invalid counter drop

        # Accept already connections
        ip protocol tcp ct state established,related counter accept
        ip protocol udp ct state established,related counter accept
        ip protocol icmp ct state established,related counter accept

        ip saddr 192.168.0.201 tcp dport ssh accept comment "SSH admin access"
        ip saddr 192.168.0.201 tcp dport 80 accept comment "nextcloud"

        # Accept ping
        icmp type echo-request limit rate 1/second counter accept
```

```
    # NTP
    udp dport 123 counter accept

    # DHCP
    udp dport 67 counter accept

    # DNS
    udp dport domain counter accept comment "DNS UDP"
    tcp dport domain counter accept comment "DNS TCP"


    meta nftrace set 1
    counter drop

}

chain forward {
    type filter hook forward priority 0; policy drop;

    # Drop invalid connections
    ip protocol tcp ct state invalid counter drop
    ip protocol udp ct state invalid counter drop
    ip protocol icmp ct state invalid counter drop

    # Accept already connections
    ip protocol tcp ct state established,related counter accept
    ip protocol udp ct state established,related counter accept
    ip protocol icmp ct state established,related counter accept

    meta nftrace set 1
    counter drop
}

chain output {
    type filter hook output priority 0; policy accept;

    # Accept all localhost traffic
    oif lo accept

    ip saddr 192.168.0.212 tcp dport 3306 accept comment "MariaDB access"

    # Drop invalid connections
    ip protocol tcp ct state invalid counter drop
    ip protocol udp ct state invalid counter drop
    ip protocol icmp ct state invalid counter drop

    # Accept already connections
    ip protocol tcp ct state established,related counter accept
    ip protocol udp ct state established,related counter accept
    ip protocol icmp ct state established,related counter accept
```

```
        meta nftrace set 1

    }
}
```

4 Règles de pare-feu MariaDB

```
#!/usr/sbin/nft -f

flush ruleset

table ip juliettedb {
    chain input {
        type filter hook input priority 0; policy drop;

        # Accept all localhost traffic
        iif lo accept
        ip saddr 127.0.0.0/8 counter drop

        # Drop invalid connections
        ip protocol tcp ct state invalid counter drop
        ip protocol udp ct state invalid counter drop
        ip protocol icmp ct state invalid counter drop

        # Accept already connections
        ip protocol tcp ct state established,related counter accept
        ip protocol udp ct state established,related counter accept
        ip protocol icmp ct state established,related counter accept

        ip saddr 192.168.0.215 tcp dport ssh accept comment "SSH admin access"
        ip saddr 192.168.0.215 tcp dport 3306 accept comment "MariaDB access"


        # Accept ping
        icmp type echo-request limit rate 1/second counter accept


        # NTP
        udp dport 123 counter accept

        # DHCP
        udp dport 67 counter accept

        # DNS
        udp dport domain counter accept comment "DNS UDP"
        tcp dport domain counter accept comment "DNS TCP"


        meta nftrace set 1
        counter drop
```

```
    }

    chain forward {
        type filter hook forward priority 0; policy drop;

        # Drop invalid connections
        ip protocol tcp ct state invalid counter drop
        ip protocol udp ct state invalid counter drop
        ip protocol icmp ct state invalid counter drop

        # Accept already connections
        ip protocol tcp ct state established,related counter accept
        ip protocol udp ct state established,related counter accept
        ip protocol icmp ct state established,related counter accept

        meta nftrace set 1
        counter drop
    }

    chain output {
        type filter hook output priority 0; policy accept;

        # Accept all localhost traffic
        oif lo accept

        # Drop invalid connections
        ip protocol tcp ct state invalid counter drop
        ip protocol udp ct state invalid counter drop
        ip protocol icmp ct state invalid counter drop

        # Accept already connections
        ip protocol tcp ct state established,related counter accept
        ip protocol udp ct state established,related counter accept
        ip protocol icmp ct state established,related counter accept

        meta nftrace set 1

    }
}
```