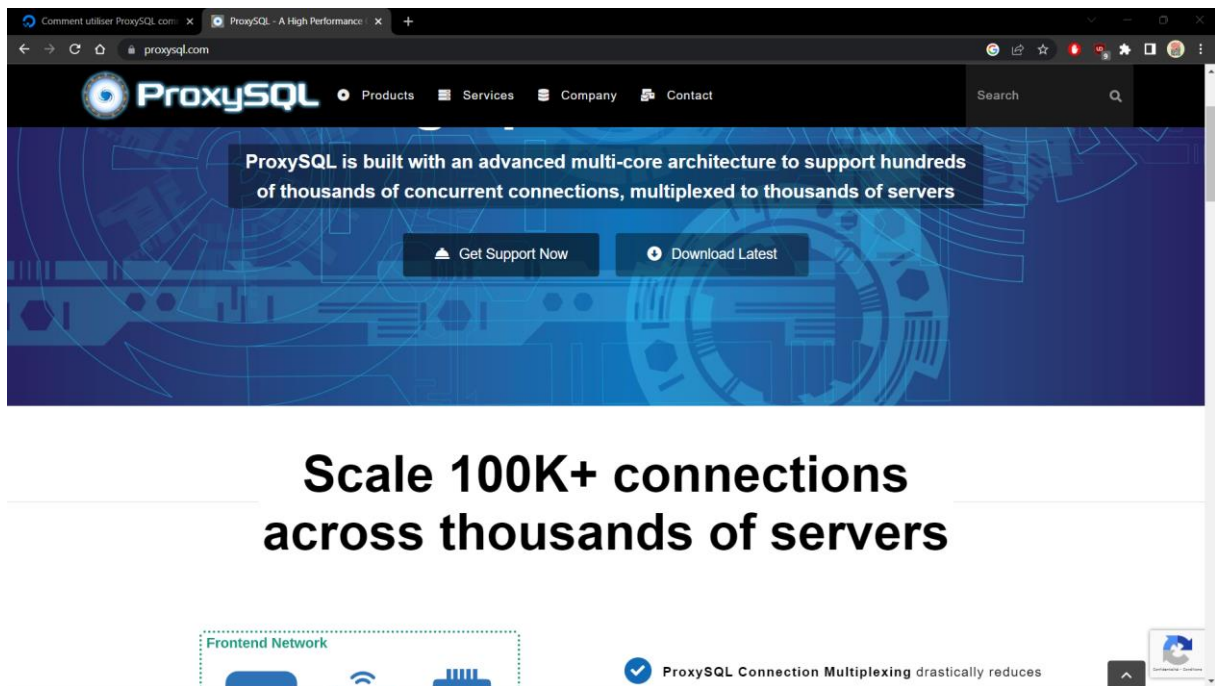


Gestion d'hôtel

Mise en place d'une base de données MYSQL répartie sur 3 nœuds en utilisant PROXYSQL

1. Installation de PROXYSQL , de MYSQL-SERVER et de MYSQL-CLIENT

Téléchargez le dernier package adéquat à la version de la distribution de notre machine virtuelle qui dans le [site proxysql.com](https://proxysql.com)



ProxySQL Core 2.5.1 AMD 64-bit Packages

ProxySQL v2.5.1 x86_64 for RedHat & CentOS 6

ProxySQL v2.5.1 x86_64 for RedHat & CentOS 7

ProxySQL v2.5.1 x86_64 for RedHat & CentOS 8

ProxySQL v2.5.1 x86_64 for Debian 8

ProxySQL v2.5.1 x86_64 for Debian 9

ProxySQL v2.5.1 x86_64 for Debian 10

ProxySQL v2.5.1 x86_64 for Debian 11

ProxySQL v2.5.1 x86_64 for Ubuntu 14

ProxySQL v2.5.1 x86_64 for Ubuntu 16

ProxySQL v2.5.1 x86_64 for Ubuntu 18

ProxySQL v2.5.1 x86_64 for Ubuntu 20

ProxySQL v2.5.1 x86_64 for Ubuntu 22

ProxySQL v2.5.1 x86_64 for AlmaLinux 8

ProxySQL v2.5.1 x86_64 for AlmaLinux 9

ProxySQL v2.5.1 x86_64 for OpenSUSE Leap 15

On utilisera la commande **dpkg -i** pour avoir le paquet sur notre machine virtuelle.

Le service proxysql ne démarre pas automatiquement après son installation, faudra donc l'activer.

```
$ sudo systemctl start proxysql
```

Output

```
• proxysql.service - LSB: High Performance Advanced Proxy for MySQL
  Loaded: loaded (/etc/init.d/proxysql; bad; vendor preset: enabled)
  Active: active (running) since Thu 2017-12-21 19:19:20 UTC; 5s ago
    Docs: man:systemd-sysv-generator(8)
  Process: 12350 ExecStart=/etc/init.d/proxysql start (code=exited, status=0/SUCCESS)
  Tasks: 23
  Memory: 30.9M
    CPU: 86ms
  CGroup: /system.slice/proxysql.service
          └─12355 proxysql -c /etc/proxysql.cnf -D /var/lib/proxysql
            └─12356 proxysql -c /etc/proxysql.cnf -D /var/lib/proxysql
```

Le **'active running'** montre que le service est bien démarré

Pour MYSQL-SERVER, on utilisera la commande **apt install mysql-server** pour l'avoir.

Ensuite, nous aurons besoin d'une application cliente MySQL pour se connecter à l'instance ProxySQL. En effet, ProxySQL utilise en interne une interface compatible avec MySQL pour les tâches administratives.

Mettez à jour votre référentiel de packages pour vous assurer que vous disposez de la dernière version pré-intégrée, puis installez le `mysql-client`.

```
$ sudo apt-get update
$ sudo apt-get install mysql-client
```

2. Définition du mot de passe de l'administrateur ProxySQL

La première fois qu'on démarre une nouvelle installation de ProxySQL, il utilise un fichier de configuration fourni par le package pour initialiser les valeurs par défaut de toutes ses variables de configuration. Après cette initialisation, ProxySQL stocke sa configuration dans une base de données qu'on peut gérer et modifier via la ligne de commande.

Pour définir le mot de passe administrateur dans ProxySQL, nous allons nous connecter à cette base de données de configuration et mettre à jour les variables appropriées.

Pour accéder à l'interface d'administration on tape la commande ci-dessous

```
$ mysql -u admin -p -h 127.0.0.1 -P 6032 --prompt='ProxySQLAdmin> '
```

Le mot de passe par défaut est `admin`

```
root@mouhamed-VirtualBox:~# mysql -u admin -p -h 127.0.0.1 -P 6032 --prompt='ProxySQLAdmin> '
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.5.30 (ProxySQL Admin Module)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

ProxySQLAdmin> UPDATE global_variables SET variable_value='admin:passer' WHERE variable_name='admin-admin_credentials';
Query OK, 1 row affected (0,00 sec)

ProxySQLAdmin> LOAD ADMIN VARIABLES TO RUNTIME;
Query OK, 0 rows affected (0,00 sec)

ProxySQLAdmin> SAVE ADMIN VARIABLES TO DISK;
Query OK, 46 rows affected (0,02 sec)

ProxySQLAdmin>
```

Ici on a modifié le mot de passe du compte administratif en mettant à jour (UPDATE) la `admin_admin_credentials` variable de configuration dans la `global_variables` base de données. Il faut aussi changer password dans la commande ci-dessous, un mot de passe fort de votre choix.

3. Configuration de la surveillance dans MySQL

ProxySQL doit communiquer avec les nœuds MySQL pour pouvoir évaluer leur état. Pour cela, il doit pouvoir se connecter à chaque serveur avec un utilisateur dédié.

Ici, nous allons configurer l'utilisateur nécessaire sur les nœuds MySQL et installer des fonctions SQL supplémentaires qui permettent à ProxySQL d'interroger l'état de réplication du groupe.

Étant donné que la réplication de groupe MySQL est déjà en cours d'exécution, les étapes suivantes ne doivent être effectuées que sur **un seul membre du groupe**

- On se connecte à MySQL, qui nous demandera à nouveau le mot de passe **root**.

```
root@mouhamed-VirtualBox:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 8.0.32-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE USER 'monitor'@'%' IDENTIFIED BY 'passer';
Query OK, 0 rows affected (0,03 sec)

mysql> GRANT SELECT on sys.* to 'monitor'@'%';
Query OK, 0 rows affected (0,01 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,02 sec)

mysql> █
```

On a créé ici l'utilisateur monitor et on lui a donné tous les privilèges pour interroger l'état du serveur MySQL

4. Configuration du monitor dans ProxySQL

```
ProxySQLAdmin> UPDATE global_variables SET variable_value='monitor' WHERE variable_name='proxyuser';
Query OK, 0 rows affected (0,00 sec)

ProxySQLAdmin> LOAD MYSQL VARIABLES TO RUNTIME;
Query OK, 0 rows affected (0,00 sec)

ProxySQLAdmin> SAVE MYSQL VARIABLES TO DISK;
Query OK, 155 rows affected (0,01 sec)

ProxySQLAdmin> █
```

Pour configurer ProxySQL pour utiliser le nouveau compte d'utilisateur lors de la surveillance des nœuds, nous mettrons à jour la variable de configuration appropriée.

5. Ajout de nœuds MySQL au pool de serveurs ProxySQL

Pour que ProxySQL connaisse nos trois nœuds MySQL, nous devons indiquer à ProxySQL comment les répartir sur ses *groupes d'hôtes*, qui sont des ensembles de nœuds désignés. Chaque groupe d'hôtes est identifié par un nombre positif, comme 1 ou 2. Les groupes d'hôtes peuvent acheminer différentes requêtes SQL vers différents ensembles d'hôtes lors de l'utilisation du routage de requête ProxySQL.

Dans les configurations de réplication statique, les groupes d'hôtes peuvent être définis arbitrairement. Cependant, la prise en charge de la réplication de groupe de ProxySQL divise automatiquement tous les nœuds d'un groupe de réplication en quatre états logiques :

- **writers**, qui sont des nœuds MySQL pouvant accepter des requêtes qui modifient les données. ProxySQL s'assure de maintenir tous les nœuds principaux jusqu'à la quantité maximale définie dans ce groupe.
- **les rédacteurs de sauvegarde**, qui sont également des nœuds MySQL qui peuvent accepter des requêtes qui modifient les données. Cependant, ces nœuds ne sont pas désignés comme écrivains ; les nœuds principaux dépassant le nombre défini d'enregistreurs maintenus sont conservés dans ce groupe et sont promus en enregistreurs si l'un des enregistreurs échoue.
- **Les lecteurs** sont des nœuds MySQL qui ne peuvent pas accepter les requêtes modifiant les données et doivent être utilisés comme nœuds en lecture seule. ProxySQL ne place ici que des nœuds esclaves.
- **offline**, qui concerne les nœuds qui se comportent mal en raison de problèmes tels qu'un manque de connectivité ou un trafic lent.

Chacun de ces quatre états a des groupes d'hôtes correspondants, mais les identificateurs de groupe numériques ne sont pas attribués automatiquement.

En mettant tout cela ensemble, nous devons dire à ProxySQL quels identifiants il doit utiliser pour chaque état. Ici, nous utilisons 1 pour le groupe d'hôtes **hors ligne** 2, pour le groupe d'hôtes d'écriture 3, pour le groupe d'hôtes **de lecture** et 4 pour le groupe d'hôtes d'écriture de sauvegarde

Voici les variables supplémentaires définies dans cette ligne et ce que chacune fait :

INSERT INTO mysql_group_replication_hostgroups (writer_hostgroup, backup_writer_hostgroup, reader_hostgroup, offline_hostgroup, active, max_writers, writer_is_also_reader, max_transactions_behind) VALUES (2, 4, 3, 1, 1, 3, 1, 100);

- **active** défini pour 1 active la surveillance par ProxySQL de ces groupes d'hôtes.
- **max_writers** définit combien de nœuds peuvent agir en tant qu'écrivains. Nous avons utilisé 3 ici parce que dans une configuration multi-primaire, tous les nœuds peuvent être traités de la même manière, nous avons donc utilisé ici 3 (le nombre total de nœuds).
- **writer_is_also_reader_set** to 1 demande à ProxySQL de traiter également les rédacteurs comme des lecteurs.
- **max_transactions_behind** définit le nombre maximum de transactions retardées avant qu'un nœud ne soit classé comme étant **hors ligne**.

Maintenant que ProxySQL sait comment répartir les nœuds entre les groupes d'hôtes, nous pouvons ajouter nos serveurs MySQL au pool. Pour ce faire, nous avons besoin de **INSERT** l'adresse IP et du groupe d'hôtes initial de chaque serveur dans la **mysql_servers** table, qui contient la liste des serveurs avec lesquels ProxySQL peut interagir.

```
ProxySQLAdmin> INSERT INTO mysql_group_replication_hostgroups (writer_hostgroup, backup_writer_hostgroup, reader_hostgroup, offline_hostgroup, active, max_writers, writer_is_also_reader, max_transactions_behind) VALUES (2, 4, 3, 1, 1, 3, 1, 100);
Query OK, 1 row affected (0,00 sec)

ProxySQLAdmin> INSERT INTO mysql_servers(hostgroup_id, hostname, port) VALUES (2, 'node1', 3306);
Query OK, 1 row affected (0,00 sec)

ProxySQLAdmin> INSERT INTO mysql_servers(hostgroup_id, hostname, port) VALUES (2, 'node2', 3306);
Query OK, 1 row affected (0,00 sec)

ProxySQLAdmin> INSERT INTO mysql_servers(hostgroup_id, hostname, port) VALUES (2, 'node3', 3306);
Query OK, 1 row affected (0,00 sec)

ProxySQLAdmin> LOAD MYSQL SERVERS TO RUNTIME;
Query OK, 0 rows affected (0,00 sec)
```

Ici, la 2 valeur définit initialement tous ces nœuds comme écrivains et 3306 définit le port MySQL par défaut.

ProxySQL devrait maintenant distribuer nos nœuds sur les groupes d'hôtes comme spécifié. Vérifions cela en exécutant une `SELECT` requête sur la `runtime_mysql_servers` table, qui expose l'état actuel des serveurs que ProxySQL utilise

```
ProxySQLAdmin> SELECT hostgroup_id, hostname, status FROM runtime_mysql_servers;
+-----+-----+-----+
| hostgroup_id | hostname | status |
+-----+-----+-----+
| 1            | node1    | ONLINE |
| 1            | node2    | ONLINE |
| 1            | node3    | ONLINE |
+-----+-----+-----+
3 rows in set (0,00 sec)
```

Tous les nœuds sont marqués `ONLINE`, ce qui signifie qu'ils sont prêts à être utilisés

6. Création des utilisateurs MySQL

ProxySQL agit comme un équilibreur de charge ; les utilisateurs finaux se connectent à ProxySQL, et ProxySQL transmet à son tour la connexion au nœud MySQL choisi. Pour se connecter à un nœud individuel, ProxySQL réutilise les informations d'identification avec lesquelles il a été accédé.

Pour autoriser l'accès aux bases de données situées sur les nœuds de réplication, nous devons créer un compte utilisateur avec les mêmes informations d'identification que ProxySQL et accorder à cet utilisateur les privilèges nécessaires. les étapes suivantes ne doivent être effectuées que sur **un seul membre du groupe** . Vous pouvez choisir n'importe quel membre.

On va créer un nouvel utilisateur appelé **hoteluser** identifié avec le mot de passe `passer` lui donné tous les privilèges sur la base de données

```
mysql> CREATE USER 'hoteluser'@'%' IDENTIFIED BY 'passer';
Query OK, 0 rows affected (0,02 sec)

mysql> GRANT ALL PRIVILEGES on hotel.* to 'hoteluser'@'%';
Query OK, 0 rows affected (0,02 sec)
```

On peut vérifier que l'utilisateur a été correctement créé en essayant d'accéder à la base de données avec les informations d'identification nouvellement configurées directement sur le nœud

```

mysql> CREATE USER 'hoteluser'@'%' IDENTIFIED BY 'passer';
Query OK, 0 rows affected (0,02 sec)

mysql> GRANT ALL PRIVILEGES on hotel.* to 'hoteluser'@'%;
Query OK, 0 rows affected (0,02 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,03 sec)

mysql> EXIT;
Bye
root@mouhamed-VirtualBox:~# mysql -u hoteluser -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 19
Server version: 8.0.32-0ubuntu0.20.04.2 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> SHOW TABLES FROM hotel;
+-----+
| Tables_in_hotel |
+-----+
| Chambres         |
| Clients          |
| Factures         |
| Reservations     |
| Services         |
+-----+
5 rows in set (0,01 sec)

mysql>

```

La liste visible des tables dans la base de données montrant la table créée dans la réplique d'origine confirme que l'utilisateur a été créé correctement sur les nœuds.

Étape 7 - Création de l'utilisateur ProxySQL

La dernière étape de configuration consiste à autoriser les connexions à ProxySQL avec l'utilisateur **hoteluser** et à transmettre ces connexions aux nœuds.

Pour ce faire, nous devons définir des variables de configuration dans la `mysql_users` table, qui contient les informations d'identification de l'utilisateur. Dans l'interface ProxySQL, on ajoute le nom d'utilisateur, le mot de passe et le groupe d'hôtes par défaut à la base de données de configuration (qui est 2, pour le groupe d'hôtes **de l'écrivain**)

```

ProxySQLAdmin> INSERT INTO mysql_users(username, password, default_hostgroup) VALUES ('hoteluser', 'passer', 2);
Query OK, 1 row affected (0,00 sec)

```

migrez la configuration dans **l'environnement d'exécution** et enregistrez-la sur **le disque** pour appliquer la nouvelle configuration.

```
ProxySQLAdmin> LOAD MYSQL USERS TO RUNTIME;  
ProxySQLAdmin> SAVE MYSQL USERS TO DISK;
```

ProxySQL écoute sur le port 6033 les connexions client entrantes, essayez donc de vous connecter à la vraie base de données (pas à l'interface d'administration) en utilisant **hoteluser** et port 6033. Vous serez invité à entrer le mot de passe, qui était **passer** dans notre exemple

```
mouhamed@mouhamed-VirtualBox:~$ sudo mysql -u hoteluser -p -h 127.0.0.1 -P 6033 --prompt='ProxySQLClient> '  
[sudo] Mot de passe de mouhamed :  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 2  
Server version: 5.5.30 (ProxySQL)  
  
Copyright (c) 2000, 2023, Oracle and/or its affiliates.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
ProxySQLClient>
```

Exécutons une instruction simple pour vérifier si ProxySQL se connectera à l'un des nœuds. Cette commande interroge la base de données sur le nom d'hôte du serveur sur lequel elle s'exécute et renvoie le nom d'hôte du serveur comme seule sortie.

```
ProxySQLClient> SELECT @@hostname;
```

Selon notre configuration, cette requête doit être dirigée par ProxySQL vers l'un de nos trois nœuds affectés au groupe d'hôtes **du rédacteur**. La sortie devrait ressembler à ce qui suit, où **node1** est le nom d'hôte de l'un des nœuds MySQL

```
ProxySQLClient> Output  
-> +-----+  
-> | @@hostname |  
-> +-----+  
-> | node1      |  
-> +-----+  
-> 1 row in set (0.00 sec)
```

Ceci termine la configuration permettant à ProxySQL d'équilibrer la charge des connexions entre les trois nœuds MySQL.